AD-A136 580

AD A-136 580

DRSMC/SA/MR-4

MODELS OF A SERVICE SYSTEM FOR PRODUCTION MACHINE MAINTENANCE

GEORGE J. SCHLENKER

NOVEMBER 1983

Distribution Statement

Approved for public release; distribution unlimited.

US ARMY ARMAMENT, MUNITIONS AND CHEMICAL COMMAND
SYSTEMS AND OPERATIONS ANALYSIS DIRECTORATE
ROCK ISLAND, ILLINOIS 61299

DISPOSITION

Destroy this report when no longer needed. Do not return it to the originator.

DISCLAIMER

The findings in this report are not to be construed as an official Department of the Army position.

WARNING

Information and data contained in this document are based on input available at the time of preparation. Because the results may be subject to change, this document should not be construed to represent the official position of the US Army Materiel Development & Readiness Command unless so stated.

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER DRSMC/SA/MR-4	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Models of a Service System for Production Machine Maintenance		5. TYPE OF REPORT & PERIOD COVERED Report-Final 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) George J. Schlenker		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS US Army Armament, Munitions and Chemical Command Systems and Operations Analysis Directorate Rock Island, IL 61299		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Armament, Munitions and Chemical Command Systems and Operations Analysis Directorate Rock Island, IL 61299		12. REPORT DATE November 1983 13. NUMBER OF PAGES 95
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Additions to/from the DISTRIBUTION LIST are invited and should be forwarded to the following address: Commander, US Army Armament, Munitions and Chemical Command, ATTN: DRSMC-SAS (R) Rock Island, IL 61299 AUTOVON 793-5041/6370

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Operations research	Maintenance systems
Markov processes	Manufacturing operations
Queueing theory	Industrial operations
Service systems	Numerical analysis

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The machine repair and adjustment process is one of the elements which characterizes the operations of a production system. In many cases this maintenance system can be described adequately by analytic models. This report derives some mathematical models which depict the steady-state performance of this service system. All the models are steady-state Markov models of a service system having a finite customer population with multiple, independent servers, each having a service time which is Erlang-distributed.

20. ABSTRACT (Cont'd)

These models have been useful in verifying the maintenance portion of a stochastic production simulation. Some numerical results from the implementing computer program are presented. Results of different models are compared. Computer programs are provided.

DRSMC/SA/MR-4

MODELS OF A SERVICE SYSTEM FOR PRODUCTION MACHINE MAINTENANCE

GEORGE J. SCHLENKER

November 1983

ACKNOWLEDGEMENT

The author gratefully acknowledges the typing of this report by Mrs. Deborah Marland. Her assistance on this and other technical reports is certainly appreciated.

CONTENTS

DRSMC-SAS (R)

November 1983

MEMORANDUM REPORT

SUBJECT: Models of a Service System for Production Machine Maintenance

1. Reference:

References are designated by bracketed numbers and are included in the footnotes. The references are also listed.

2. Background

This memorandum reports on work associated with a larger study [1], concerned with the productivity of manufacturing systems. A stochastic simulation, called TANDEMT, was developed for the Ref [1] study. Part of TANDEMT characterizes a maintenance system for adjusting and repairing machines on a production line. To verify this part of the simulation, analytic studies of service systems were performed. The models in this study draw from and extend classical queueing theory, such as in Ref [2]. Whereas the analytic models were used to verify a simulation, their use is not limited to that purpose. It has been found that one can forego the simulation and use only analytic models in designing a machine maintenance system*. Recognizing

---

[1] DD1498, HQ, US ARRCOM, DRSAR-SA, March 1983, title: Manufacturing Productivity Study.

[2] Gross, Donald and Harris, Carl, Fundamentals of Queueing Theory, John Wiley and Sons, c. 1974.

* The analytic models are included in a subroutine which can be called from TANDEMT. This arrangement is convenient since these programs share input data. In using the routine for system design, the minimum number of repairmen is selected which satisfies constraints on system performance measures.

the general* importance of analytic models of this sort, I have chosen to present certain derivations and results of interest here.

3. Objectives

One purpose of this memorandum is to derive some results which may be useful in modeling a maintenance system which serves a finite population of machines. Another objective is to present parametric studies using these models. A comparison of results of alternative models will also be made. Of interest to analysts are some numerical approximations.

4. Scope

In the language of queueing theory the (finite) population of machines are "customers" and the repairmen who perform maintenance are the "servers". Similarly, the failures of machines are regarded as demands for service, which if unmet are placed in a "service queue". I will use these terms to suggest that these models are not limited to a production setting. The "system" of interest here is the machine maintenance system, which is characterized by the number of repairmen (or servers) and by the probability distribution of service time. These two factors are the distinguishing elements between models in this memorandum. Note that no distinction is made between types of repairmen. For this application any repairman can render the requisite service. These repairmen are not specialists. Further, they are assumed to work as individuals, rather than in teams.

---

* Other applications of the models considered here are manifold. Finite queueing models have been used in describing the process by which artillery targets are detected and "served" and in describing the FM communications process in tactical fire control nets.

5. Ordinarily, the random arrival of a "demand" for service is regarded as occuring at a rate equal to the product of a unit rate times the number of customers outside* the service system. The time-to-next-demand is conditionally an exponential random variable. For most mature manufacturing systems operating in steady-state, this is a valid description of demand for maintenance. This demand description is used in all models in this report. Actual maintenance experience often shows that the time to complete a maintenance action (time to repair) is a lognormal random variable. To describe a lognormally distributed event in terms of Markov processes involves approximation. For service times which have a coefficient of variation near unity, an exponential approximation is used. However, if the standard deviation of service time is significantly smaller than the mean--as is often the case--an Erlang distribution can be used to describe the service time. The Erlang distribution, or gamma distribution with integer shape parameter p can be represented as the distribution of a sum of p random variables each of which is exponential. For example, in the case of an Erlang distribution with shape parameter 2 and mean $\overline{x}$, an equivalent stochastic model consists of two serial elements both of which are exponential with common mean $\overline{x}/2$. Generally, the coefficient of variation of a gamma distribution with shape parameter p is $1/\sqrt{p}$. Thus, <u>if</u> the standard deviation of service time is about 0.7 times the mean, an Erlang distribution with shape parameter 2 would be an appropriate statistical model. The models considered here have either exponential or gamma(2) service times.

6. <u>Birth-Death Processes</u>

Consider a service system having a <u>single</u> server whose service times are gamma(2) i.e., are random variables from a gamma probability distribution with shape parameter 2. This is sometimes written $\Gamma(2)$. Let the customer population be m machines. Thus, at most m machines may reside in the maintenance system with at most m-1 in the queue. Machine failures occur at inter-event times which are exponential with a rate $(m-k)\lambda$, where k machines are in the system and where

---

* The term "inside the system" refers to residence in either the service queue or in a service channel being served. After being served a customer leaves the system and is then "outside the system".

the unit rate λ is the unit rate of entry or "birth" rate for a machine into the system. Notationally, let MTBF be the mean time between failures for the population of machines.* Then,

$$\lambda = 1/\text{MTBF}. \tag{1}$$

To model the gamma(2) service times, two stages of service are assumed, each of which is exponentially distributed with rate parameter μ. After the second stage of service is completed the machine leaves the maintenance system. Thus, if μ_1 is the rate of exodus, or "death" rate, from the system

$$\mu = 2\mu_1 . \tag{2}$$

With a mean time to repair MTTR,

$$\mu_1 = 1/\text{MTTR} . \tag{3}$$

7. Definitions of States for a Single-Server System

To characterize the possible states of the system, define and number event E_j as follows:

$$E_j = 2j-1 \quad , \qquad j \geq 1 \quad , \tag{4a}$$

with j-1 customers in the queue, where the customer being served is in the first stage of service; and

$$E_j = 2j \quad , \tag{4b}$$

with j-1 customers in the queue, where the customer being served is in the second stage of service. E_0 denotes the null state. Notationally, let

$$p_n(t) = P\{\text{system is in state } E_n \text{ at time } t\}, \quad 0 \leq n \leq 2m . \tag{5}$$

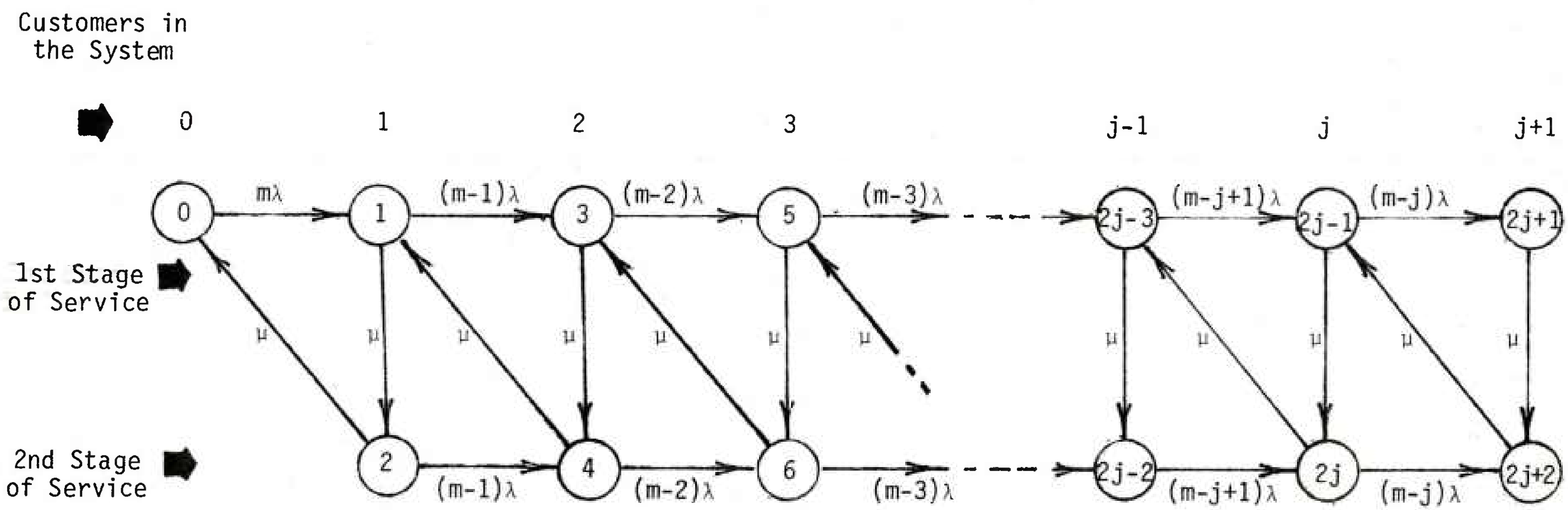
The object of the following derivation is to show how $p_n(t)$ is obtained, for the admissible values of n, when time becomes large, i.e., when the system is in stochastic steady state.

8. State Transitions (Single Server)

It is helpful to examine the ways in which the system can transition between states. Figure 1 displays the states as nodes and the transitions as directed arcs. This state transition diagram also shows the Markov rate constant for

---

* See Annex B for treatment of a heterogeneous population of machines.



5

Definition of States

State E_j is defined by the number of customers in the system and by the stage of service as follows:

E_j = 2j-1 with j customers in the system and with the customer being attended in the 1st stage of service.

= 2j with j customers in the system and with the customer being attended in the 2nd stage of service,

$1 \leq j \leq m$.

Figure 1. State Transition Diagram for a Single-Server Service System With Erlang (2) Service Times

each indicated transition as a label on the arc. Thus, the conditional probability of transition from state 0 to state 1 in an incremental time interval h is $m\lambda h$. Similarly, given the system is in state 1, a transition to state 2 in time increment h occurs with probability μh, and a transition to state 3 in time increment h occurs with probability $(m-1)\lambda h$. Following the standard procedure for Markov processes, one writes the probability of being in state n at time t+h in terms of the probabilities of occupying all of the states at time t. For example,

$$\begin{aligned} P\{E_0 \text{ at } t+h\} &= P\{E_0 \text{ at } t\} \cdot \\ &P\{\text{no transition in } (t, t+h)\} \\ &+ P\{E_2 \text{ at } t\}P\{\text{trans. from } E_2 \text{ to } E_0 \text{ in } (t, t+h)\} . \end{aligned} \tag{6}$$

In our notation,

$$p_0(t+h) = p_0(t)\ (1 - m\lambda h) + p_2(t)\mu h . \tag{7}$$

Then,

$$\frac{1}{h}\,[p_0(t+h) - p_0(t)] = -m\lambda p_0(t) + \mu p_2(t) . \tag{8}$$

Taking the limit as h approaches zero yields

$$\dot{p}_0(t) = -m\lambda p_0(t) + \mu p_2(t) . \tag{9}$$

Note that this result can also be obtained by inspection from the state transition diagram. To form the general right hand side: All rates from the k th node are summed; and, the negative of this sum is the coefficient of p_k. All other terms of the r.h.s. have positive coefficients corresponding to the rates on arcs entering the k th node. The j th coefficient multiplies $p_j(t)$, where j is a node leading directly into the k th node.

9. Kolmogorov Equations for a One-Server System

Following the above procedure for all states gives rise to the differential-difference (Kolmogorov) equations which describe this system. Suppressing the notation for time dependence, one has:

$$\dot{p}_0 = -m\lambda p_0 + \mu p_2 .$$

$$\dot{p}_1 = m\lambda p_0 - (\mu+(m-1)\lambda)p_1 + \mu p_4 .$$

For $2 \leq j \leq m-1$,

$$\dot{p}_{2j-1} = (m-j+1)\lambda\, p_{2j-3} - (\mu+(m-j)\lambda)p_{2j-1} + \mu p_{2j+2}$$

$$\dot{p}_{2j} = (m-j+1)\lambda\, p_{2j-2} + \mu p_{2j-1} - (\mu+(m-j)\lambda)p_{2j} .$$

And, for the last two states,

$$\dot{p}_{2m-1} = \lambda p_{2m-3} - \mu p_{2m-1}$$

$$\dot{p}_{2m} = \lambda p_{2m-2} + \mu p_{2m-1} - \mu p_{2m} . \tag{10}$$

In stochastic steady state, the time derivatives of all probabilities are zero. Setting dp_k/dt to zero for all k in (10) yields a matrix-vector equation for the state probability vector $\underline{p}$, with

$$\underline{p}' = (p_1, p_2, \ldots, p_{2m}) .$$

The 2m+1 equations in the set (10) contain one redundant equation since the probabilities are related by

$$p_0 = 1 - \Sigma_{j=1}^{2m} p_j . \tag{11}$$

One of the equations can be deleted--say, the first--and (11) substituted for p_0 in the second equation (the only other place in which p_0 appears). This process yields

$$-m\lambda\Sigma_{j=1}^{2m} p_j - (\mu + (m-1)\lambda)p_1 + \mu p_4 = -m\lambda$$

$$(m-1)\lambda p_1 - (\mu + (m-2)\lambda)p_3 + \mu p_6 = 0$$

$$(m-1)\lambda p_2 + \mu p_3 - (\mu + (m-2)\lambda)p_4 = 0 ,$$

and so forth.

The equivalent matrix-vector formulation is

$$A\, \underline{p} = \underline{b} \tag{12}$$

with A (2m x 2m) and $\underline{p}$ and $\underline{b}$ (2m x 1), where

$$\underline{b}' = (-m\lambda, 0, 0, \ldots, 0) , \tag{12b}$$

and

$$A = \{a_{ij}\} . \tag{12c}$$

The elements of A are:

$$a_{11} = -m\lambda - (\mu + (m-1\lambda)$$

$$a_{1j} = -m\lambda \quad , \quad 2 \leq j \neq 4 \leq 2m$$

$$a_{14} = \mu - m\lambda$$

$$a_{21} = \mu$$

$$a_{22} = -(\mu + (m-1)\lambda)$$

$$a_{2j} = 0 \quad , \quad j > 2 .$$

For the remaining non-zero terms

of $\{a_{ij}\}$, $2 \leq k \leq m-1$,

$$a_{2k-1,\ 2k-3} = (m-k+1)\lambda$$

$$a_{2k-1,\ 2k-1} = -\ (\mu+(m-k)\lambda)$$

$$a_{2k-1,\ 2k+2} = \mu$$

$$a_{2k,\ 2k-2} = (m-k+1)\lambda$$

$$a_{2k,\ 2k-1} = \mu$$

$$a_{2k,\ 2k} = -\ (\mu+(m-k)\lambda) \ ;$$

$$a_{2m-1,\ 2m-3} = \lambda$$

$$a_{2m-1,\ 2m-1} = -\mu$$

$$a_{2m,\ 2m-2} = \lambda$$

$$a_{2m,\ 2m-1} = \mu$$

$$a_{2m,\ 2m} = -\mu \ . \tag{13}$$

Since A is of full rank, $\underline{p}$ may be obtained from

$$\underline{p} = A^{-1}\underline{b} . \tag{14}$$

The probability of the null state is obtained from (11).
Then, the probability of the maintenance system having k customers in steady state is given by

$$\pi_k = p_{2k-1} + p_{2k} \quad , \quad 1 \leq k \leq m . \tag{15}$$

10. Statistical Properties of the System

From this probability distribution one obtains the mean and variance* of the number of customers in the system.

$$E[N_{sys}] \equiv \overline{N}_{sys} = \Sigma_{k=1}^{m} k\pi_k \tag{16}$$

$$V(N_{sys}) = \Sigma_{k=1}^{m} k^2 \pi_k - (\overline{N}_{sys})^2 . \tag{17}$$

The (mathematically) expected number of customers in the queue ($\overline{N}_q$) is obtained from

$$E[N_q] = \Sigma_{k=2}^{m} (k-1)\pi_k$$

or

$$E[N_q] = \overline{N}_{sys} - (1-p_0) \equiv \overline{N}_q \tag{18}$$

The variance of the number in the queue

$$V[N_q] = \Sigma_{k=2}^{m} (k-1)^2\pi_k - (\overline{N}_q)^2 . \tag{19}$$

Related properties are the mean time a customer spends in the system (W) and the mean time spent in the queue (W_q). For all service systems the value of W is greater than W_q by the mean service time. In this case,

$$W = W_q + 1/\mu_1 . \tag{20}$$

---

* Operator notation is used in displaying the mean and variance. The expectation operator is denoted E[·], and the variance operator V[·].

Given W, equation (20) is used to obtain W_q.

With exponential arrivals and a finite customer population the mean arrival rate is $\lambda(m-\overline{N}_{sys})$. Using this result in Little's formula (p. 120, [2]) yields

$$\overline{N}_{sys} = \lambda(m-\overline{N}_{sys})W ,$$

from which

$$W = \overline{N}_{sys}/[\lambda(m-\overline{N}_{sys})] . \tag{21}$$

Then, W_q can be obtained from (20). Alternatively, W_q is given by another version of Little's formula:

$$W_q = \overline{N}_q/[\lambda(m-\overline{N}_{sys})] . \tag{22}$$

11. Conditional State Probability Distribution

The following derivation is to obtain the probability distributions of waiting time in queue and in the system, given an arriving customer must queue. In doing this, we first obtain the conditional probability distribution that the system is in the n th state, given an arrival. Denote, q_n as the (steady-state) probability that an arrival finds the system in the n th state. From Bayes' theorem, q_n is proportional to the product of two probabilities: (a) P{that an arrival occurs, given that the system is in state n} and (b) P{system is in state n}. Thus,

$$\begin{aligned}
q_0 &\propto m\lambda\ p_0 \\
q_1 &\propto (m-1)\lambda\ p_1 \\
q_2 &\propto (m-1)\lambda\ p_2 \\
q_3 &\propto (m-2)\lambda\ p_3 \\
q_4 &\propto (m-2)\lambda\ p_4 \\
&\vdots \\
q_{2m-3} &\propto (m-1)\lambda\ p_{2m-3} \\
q_{2m-2} &\propto (m-1)\lambda\ p_{2m-2} \cdot
\end{aligned} \tag{23}$$

[2] Gross, D. and Harris C. Op. Cit., 1974.

Generally, denoting the integer portion of x by int[x] ,

$$q_n = (m-\text{int}[(n+1)/2])p_n/K \quad , \qquad 0 \le n \le 2m-2 \ , \tag{24}$$

where the normalization constant K is given by

$$K = mp_0 + \Sigma_{n=1}^{2m-2} (m - \text{int}[(n+1)/2])p_n \tag{25}$$

The conditional probability density that an arrival finds k other customers in the system is denoted by ρ_k:

$$\rho_0 = q_0$$

$$\rho_k = q_{2k-1} + q_{2k} \quad , \qquad 1 \le k \le m-1 \ . \tag{26}$$

This function is analogous to the unconditional probability density π_k, given by equation (15).

12. Notationally, denote $F_q(t)$ as the distribution of waiting time in queue, T_q:

$$F_q(t) = P\{T_q \le t\} \quad . \tag{27}$$

The expected value of time in queue from this distribution has been denoted above as W_q. Denote the conditional expectation of time in queue, given a queue as $W_{q|w}$. For a single-server system, queueing occurs whenever the system is not in the null state. Therefore,

$$W_{q|w} = W_q/(1-q_0) \ . \tag{28}$$

An alternative -- to equation (22)--means of calculating W_q in this case is the following. This expression is based on the fact--as observed in the state transition diagram (Figure 1)--that an odd state with k customers (E_{2k-1}) involves 2k stage transitions of service to leave the system; and, an even state with k customers (E_{2k}) involves 2k-1 stage transitions of service to leave the system. Each service-stage transition requires an expected time of $1/\mu$. Thus,

$$W_q = \frac{1}{\mu} \Sigma_{k=1}^{m-1} (2k \ q_{2k-1} + (2k-1)q_{2k}) \ . \tag{29}$$

Altho it is not obvious in this case, the mean of waiting time in queue given by (29) equals the value given by the expression (22) from Little's formula. Clearly, equation (22) is preferable computationally; however, development of (22) has motivated the method for deriving $F_q(t)$.

13. Define an integer n* such that

$$n^* = n+1 \quad \text{for } n \text{ odd}$$

$$n^* = n-1 \quad \text{for } n \text{ even} , \tag{30}$$

with n an index over the system states: $1 \leq n \leq 2m-2$.

Then,

$$F_q(t) = q_0 + \Sigma_{n=1}^{2m-2} q_n P\{n^* \text{ stages of service are completed in time } \leq t, \text{ given an arrival finds the system in the n th state}\} . \tag{31}$$

Since each stage of service requires an exponentially-distributed time, with mean μ^{-1}, the time to complete n* stages has a cumulative distribution which is the n*-fold convolution of this, namely the gamma cumulative distribution function (c.d.f.):

$$G(n^*,\mu,t) = \int_0^t \frac{\mu(\mu x)^{n^*-1}}{(n^*-1)!} e^{-\mu x} dx . \tag{32}$$

Using (32), one can write (31) as

$$F_q(t) = q_0 + \Sigma_{n=1}^{2m-1} q_n G(n^*,\mu,t) . \tag{33}$$

With a change in the summation index and using (30), equation (33) becomes

$$F_q(t) = q_0 + \Sigma_{k=1}^{m-1} (q_{2k-1} G(2k,\mu,t) + q_{2k} G(2k-1,\mu,t)) . \tag{34}$$

A computationally useful form of (34) may be derived using Molina's theorem ([3] and [4], p. 86):

$$\int_0^x \frac{u^{n-1} e^{-u}}{(n-1)!} du = 1 - \Sigma_{j=0}^{n-1} \frac{x^j e^{-x}}{j!} ; \tag{35a}$$

$$= G(n,1,x) . \tag{35b}$$

---

[3] Molina, E.C. Poisson's Exponential Binomial Limit, Van Nostrand, c. 1942.

[4] Schlenker, G. Numerical Methods in Renewal Theory, (AD 828276), Hdqts USA WECOM, February 1968.

Molina's theorem relates a standardized gamma c.d.f. to a sum of Poisson terms. Note that the c.d.f. in (32) with parameters n* and μ and with argument t is, via a change of variable, equivalent to $G(n^*,1,\mu t)$. Thus, the gamma c.d.f.'s in (34) can be replaced by the sums required by (35) using an argument x, where

$$x = \mu t \ . \tag{36}$$

This substitution yields the formula for calculating $F_q(t)$:

$$F_q(t) = 1 - \Sigma_{k=1}^{m-1} \left(q_{2k-1} \Sigma_{j=0}^{2k-1} \frac{x^j e^{-x}}{j!} + q_{2k} \Sigma_{j=0}^{2k-2} \frac{x^j e^{-x}}{j!}\right) , \tag{37}$$

or

$$F_q(t) = 1 - q_1 (e^{-x} + xe^{-x}) - q_2 e^{-x}$$
$$- e^{-x} \Sigma_{k=2}^{m-1} \left(q_{2k-1} \Sigma_{j=0}^{2k-1} \frac{x^j}{j!} + q_{2k} \Sigma_{j=0}^{2k-2} \frac{x^j}{j!}\right) . \tag{38}$$

This unconditional probability distribution for time in queue can also be used to get the conditional c.d.f., given an arrival must queue. The latter is denoted by $F_{q|w}(t)$. Since q_0 is the probability that an arrival does not need to queue,

$$F_{q|w}(t) = (F_q(t) - q_0)/(1-q_0) \ . \tag{39}$$

Using (34) and (39),

$$F_{q|w}(t) = \frac{1}{1-q_0} \Sigma_{k=1}^{m-1} (q_{2k-1} G(2k,\mu,t) + q_{2k} G(2k-1,\mu,t)) \ . \tag{40}$$

Note that the conditional c.d.f. of time in queue is a weighted sum of gamma distributions.

14. Mean and Variance of Conditional Waiting Time in Queue

The j th origin moment of $G(n,\mu,t)$ is given, from p. 33, [4], as

$$a_j = \mu^{-j} \Pi_{i=1}^{j} (n+i-1) \ . \tag{41}$$

---

[4] Schlenker, G. Op. Cit.

Specifically, the first and second origin moments are:

$$a_1 = n\mu^{-1}$$

$$a_2 = n(n+1)\mu^{-2} \; . \qquad (42)$$

Using these results, one can obtain expressions for the first and second origin moments for the conditional time in queue ($T_{q|w}$).
From (40),

$$E[T_{q|w}] = \frac{1}{1-q_0} \; \Sigma_{k=1}^{m-1} \; q_{2k-1} \; 2k\mu^{-1} + q_{2k} \; (2k-1)\mu^{-1} \; , \qquad (43)$$

and

$$E[T^2_{q|w}] = \frac{1}{1-q_0} \; \Sigma_{k=1}^{m-1} \; q_{2k-1} \; 2k(2k+1)\mu^{-2} + q_{2k} \; 2k(2k-1)\mu^{-2} \; . \qquad (44)$$

Then, the variance of the conditional time in queue is obtained from

$$V[T_{q|w}] = E[T^2_{q|w}] - (E[T_{q|w}])^2 \; . \qquad (45)$$

The expression $E[T_{q|w}]$ was earlier denoted by $W_{q|w}$, and was related to W_q by (28). Thus, equation (28) together with (22) provide a consistency check for the value given by (43). This check is performed in the attached computer program FINITE.ME2.Q.

15. Multiple Servers With Exponential Service Times

The formulas derived at this point pertain to the case in which a single server exhibits service times which are Erlang(2). The Kolmogorov equations for this case, tho simple enough, did not lead to a solution of the state probabilities (p_n) in closed form. All equations involved numerical solution with a degree of complexity requiring computer assistance. Another model of a finite-customer population which does lend itself to a somewhat simpler solution is one to which I now turn. In this model the service times are exponential random variables--i.e., Erlang(1) distributed. Further, the number of servers is generalized to an arbitrary integer c. This model has been developed in some detail by Gross and Harris [2]. For this model p_n represents the probability that n customers are found in the service system in stochastic steady state. From p. 118 of Ref [2],

[2] Gross, D. and Harris, C. Op. Cit.

$$p_n = \binom{m}{n} \left(\frac{\lambda}{\mu}\right)^n p_0 \quad , \qquad 0 \leq n < c \tag{46a}$$

$$p_n = \binom{m}{n} \frac{n!}{c^{n-c}c!} \left(\frac{\lambda}{\mu}\right)^n p_0 \quad , \qquad c \leq n \leq m \; . \tag{46b}$$

The value of the probability for the null state is found by setting the sum of the state probabilities to unity.

$$p_0^{-1} = \Sigma_{n=0}^{c-1} \binom{m}{n} \left(\frac{\lambda}{\mu}\right)^n + \Sigma_{n=c}^{m} \binom{m}{n} \frac{n!}{c^{n-c}c!} \left(\frac{\lambda}{\mu}\right)^n \; . \tag{47}$$

Equations for the mean and variance of the number of customers is the system--analogous to (16) and (17)--are:

$$\overline{N}_{sys} = \Sigma_{k=1}^{m} k \, p_k \tag{48}$$

$$V(N_{sys}) = \Sigma_{k=1}^{m} k^2 \, p_k - (\overline{N}_{sys})^2. \tag{49}$$

Since queueing occurs when $N_{sys} > c$, the expected number of customers in queue is given by

$$\overline{N}_q = \Sigma_{k=c}^{m} (k-c) \, p_k$$

or

$$\overline{N}_q = \overline{N}_{sys} - c + \Sigma_{k=0}^{c-1} (c-k) \, p_k \; . \tag{50}$$

Little's formula is also valid for this model; hence, the mean system waiting time and mean waiting time in queue are given by (21) and (22), respectively.

16. Distribution of Time in Queue (Exponential Services)

The notational conventions used in pgf 11 for the model with Erlang(2) service times are also used here. The conditional probability that an arriving customer finds the system in the n th state is denoted q_n. Bayes' rule requires that

$$q_n = (m-n)p_n/K \; , \tag{51}$$

where the normalization constant K is found by requiring that

$$\Sigma_{n=0}^{m-1} q_n = 1.$$

Thus,

$$K = 1/\Sigma_{n=0}^{m-1} (m-n)p_n \; . \tag{52}$$

The following derivation for the c.d.f. of time in queue for a c-server system is a generalization of a result found on p. 125 of [2].
Let $F_q(t)$ be the cumulative (unconditional) distribution of waiting time in queue as in (27). Then,

$$F_q(t) = \Sigma_{n=c}^{m-1} q_n P\{\text{n-c+1 job completions occur in time} \le t, \text{ given an arrival finds n in the system}\} + P\{\text{system is open}\} , \tag{53a}$$

where

$$P\{\text{system is open}\} = \Sigma_{n=0}^{c-1} q_n . \tag{53b}$$

In this model a job completion occurs in exponentially distributed time with rate constant $c\mu$, when the system is full.
Therefore, the

$$P\{n^* \text{ completions occur in time} \le t\}$$

is given by

$$G(n^*, c\mu, t) ,$$

with the gamma c.d.f. defined by (32).
Thus,

$$F_q(t) = \Sigma_{n=c}^{m-1} q_n G(n-c+1, c\mu, t) + \Sigma_{n=0}^{c-1} q_n . \tag{54}$$

Using Molina's theorem (35), a computationally useful form of (54) is obtained.

$$F_q(t) = 1 - q_c e^{-c\mu t} - e^{-c\mu t} \Sigma_{n=c+1}^{m-1} q_n \Sigma_{j=0}^{n-c} \frac{(c\mu t)^j}{j!} . \tag{55}$$

From (54) it is seen that the unconditional expected time in queue is the weighted sum of the means of gamma distributions having different shape parameters. With (42),

$$E[T_q] = \Sigma_{n=c}^{m-1} q_n (n-c+1)/(c\mu) . \tag{56}$$

Similarly,

$$E[T_q^2] = \Sigma_{n=c}^{m-1} q_n (n-c+1)(n-c+2)/(c\mu)^2 . \tag{57}$$

Then, the unconditional variance of time in queue is

$$V(T_q) = E[T_q^2] - (E[T_q])^2 . \tag{58}$$

---

[2] Gross, D. and Harris, C. Op. Cit.

The conditional probability distribution of queueing time, given an arrival must wait, is obtained from $F_q(t)$ by

$$F_{q|w}(t) = (F_q(t) - P\{open\})/(1-P\{open\}) , \tag{59}$$

where P{open} is given by (53b).
This expression is analogous to (39) for the single-server model.
The first two origin moments of the conditional probability distribution of queueing time are obtained from the corresponding unconditional moments (56, 57) by dividing each by 1-P{open}. The variance of this conditional c.d.f. is obtained from the general expression in (45).

17. Numerical Results

Before proceeding with the derivation of other analytic relationships, it is interesting to apply the formulas developed at this point to specific examples. The numerical values chosen are motivated by the author's experience with maintenance of machines for producing metal parts of conventional ammunition. In performing the calculations two computer programs have been useful: FINITE.ME2.Q and BEST.SERVICE. These subprograms are listed in Annex A together with an executive driver. The first program implements the model with conditional Poisson* arrivals and Erlang(2) services. The second program calculates all the statistics for the multiserver model having conditionally Poisson arrivals and exponential services. Both programs treat the population of customers as finite. In order to contrast results from a finite-population model with a model having an infinite customer population, I also evaluate the latter model with an arrival rate parameter (λ_∞) chosen to yield the maximum arrival rate in the finite models. Thus,

$$\lambda_\infty = m\lambda . \tag{60}$$

---

* Poisson arrivals implies that the time between arriving customers is an exponentially distributed random variable and the arrival rate is a constant parameter. In this case, I mean by "conditionally Poisson" that the arrival rate per customer outside the system is a constant.

The model with infinite customer population which is chosen for comparison with the models of this report has an Erlang(2) service time distribution with a single server. Certain mean-value statistics for this case are presented in [2], starting on page 163. This model is referred to as $(M/E_2/1)$ by Gross and Harris [2].

The expected waiting time--W_q, in the notation of [2]--for $(M/E_2/1)$ is

$$W_q = \frac{3\lambda_\infty}{4\mu_\infty(\mu_\infty-\lambda_\infty)}$$

where μ_∞ is the stage service rate, denoted elsewhere as μ.
The expected waiting time in the system (W) is given by (20), since this equation is independent of population size. Little's formula yields the expected number in the queue (L_q) and the expected number in the system (L):

$$L_q = \lambda_\infty W_q \tag{62a}$$

$$L = \lambda_\infty W \,. \tag{62b}$$

The probability that the service system is empty (p_0) is also given in [2]:

$$P\{N=0\} \equiv p_0 = 1 - \lambda_\infty/\mu_\infty \,. \tag{63}$$

Equations (60) thru (63) were evaluated for comparison with their counterparts of the finite-population models. A numerical comparison of the infinite-population model with the finite population models is made in Table 1. The notational designations for the three models used in Table 1 are Infinite $\Gamma(2)$, Finite $\Gamma(2)$, and Finite Exponential to indicate the customer population size and the service-time distribution function, respectively. The time between arrivals (in minutes) and the number of customers in the population are treated as parameters. One notes that certain output variables agree fairly well across models, such as p_0, whereas others, such as W_q, do not. Due to the fact that λ_∞ exceeds the average arrival rate in the finite models, the average number in the system and in the queue exceeds their finite-population counterparts. However, it is noteworthy that when the time between arrivals is large (or λ is small) and the customer population is large, the values of E[N] and E[Q] from the infinite model approximate the corresponding values in the finite $\Gamma(2)$ model.

TABLE 1

COMPARISON OF THE STOCHASTIC PROPERTIES OF
SEVERAL MODELS OF MAINTENANCE SERVICE SYSTEMS

Assumptions:
Distribution of time between arrivals is exponential for each customer in a finite population of customers.
There is a single server with mean service time of 20 minutes.

$1/\lambda$ (1) (min)	Number Customers	System (2) Property	Type of Model (3)		
			Inf., $\Gamma(2)$	Finite, $\Gamma(2)$	Finite, Exp
120	2	E[N]	0.4583	0.3128	0.3200
		SD[N]		0.5274	0.5455
		E[Q]	0.1250	0.0316	0.0400
		P{N=0}	0.6667	0.7188	0.7200
		Wq	7.500	2.249	2.857
		Wq|W		14.800	20.000
	3	E[N]	0.8750	0.5191	0.5410
		SD[N]		0.6934	0.7372
		E[Q]	0.3750	0.1057	0.1311
		P{N=0}	0.5000	0.5865	0.5902
		Wq	15.000	5.111	6.400
		Wq|W		17.181	22.857
	5	E[N]	3.9582	1.0936	1.1624
		SD[N]		1.0479	1.1507
		E[Q]	3.1249	0.4425	0.5228
		P{N=0}	0.1667	0.3489	0.3604
		Wq	74.998	13.593	16.348
		Wq|W		24.164	30.820
	10	E[N]	∞	4.1705	4.2588
		SD[N]		1.9462	2.1538
		E[Q]		3.1989	3.3020
		P{N=0}		0.0284	0.0431
		Wq		65.849	69.017
		Wq|W		68.824	74.625
	20	E[N]	∞	14.0000	14.0000
		SD[N]		2.1354	2.4494
		E[Q]		13.0000	13.0000
		P{N=0}		0.0000	0.0000
		Wq		260.000	260.001
		Wq|W		260.000	260.004

TABLE 1 (CONT)

COMPARISON OF THE STOCHASTIC PROPERTIES OF
SEVERAL MODELS OF MAINTENANCE SERVICE SYSTEMS

Assumptions:
Distribution of time between arrivals is exponential for each customer in a finite population of customers.
There is a single server with mean service time of 20 minutes.

$1/\lambda$ (1) (min)	Number Customers	System (2) Property	Type of Model (3) Inf., $\Gamma(2)$	Finite, $\Gamma(2)$	Finite, Exp
1200	20	E[N]	0.4583	0.4336	0.4668
		SD[N]		0.7210	0.8012
		E[Q]	0.1250	0.1075	0.1413
		P{N=0}	0.6667	0.6739	0.6744
		Wq	7.500	6.591	8.679
		Wq|W		21.138	28.049
	30	E[N]	0.8750	0.7998	0.8928
		SD[N]		1.0634	1.2358
		E[Q]	0.3750	0.3131	0.4076
		P{N=0}	0.5000	0.5133	0.5149
		Wq	15.000	12.868	16.806
		Wq|W		27.186	35.809

Notes for Table 1.

(1) The mean time between arrivals per customer is $1/\lambda$. Thus, the maximum arrival rate is λ times the number of customers.

(2) The following notation is used to denote system properties:

E[N]	Statistically expected number of customers in the service system.
SD[N]	Standard deviation of the number of customers in the service system.
E[Q]	Expected value of the number in the service queue.
P{N=Ø}	Probability that the system is empty.
Wq	Expected value of the waiting time (minutes) in the service queue.
Wq|W	Expected waiting time (minutes) given a customer must queue.

(3) Models are characterized here by the population of customers and by the form of the service-time distribution. Thus, "Inf., Γ(2)" signifies an infinite population of customers and a service-time distribution which is gamma (or Erlang) with shape parameter 2. The form of the c.d.f. of time t is

$$F(t) = 1 - (1+\beta t)e^{-\beta t},$$

with parameter β. The mean of t with this distribution is $2/\beta$.

The other model has a finite customer population and exponentially distributed service times.

18. Comparisons Between Finite Models

The model with exponential service times is computationally much easier than the $\Gamma(2)$ model. Therefore, it is useful to ask for what statistics the models are in reasonable agreement. For these, the model with exponential service will do. Some observations can be made regarding the similarity of results from the finite $\Gamma(2)$ model versus the finite exponential model. Generally, very good agreement exists between these models for the values of $E[N]$, $SD[N]$, and $P\{N=0\}$. Poorer agreement exists for $E[Q]$, and very poor agreement exists for W_q and $W_{q|w}$. Clearly, the difference in dispersion of service times has the greatest effect upon mean waiting time (or conditional mean waiting time). Differences in the conditional waiting time distributions are displayed explicitly in Figures 2 and 3. These figures differ in only one parameter--customer population size. Figures 4 and 5 illustrate the parametric effect of population size for the model with exponential services. Using this model, the parametric effect of number of servers on the c.d.f. of waiting time in queue is shown in Figure 6. These probability distributions are plotted on Weibull probability paper for convenience, since extreme variation in waiting time is shown for any value of a parameter. The following interesting observation can be made from Figure 6. When the number of servers increases--indicating improving service--the conditional distribution function of time in queue closely approximates an exponential distribution. An exponential distribution plots as a straight line on Weibull paper with a slope of unity (Note scale difference of abscissa and ordinate.) Even tho the mathematical form of (55) is clearly not that of an exponential c.d.f., nevertheless an exponential form is a good approximation under certain conditions. After examining a variety of parametric variations with this model, I conclude that a condition under which the exponential approximation is pragmatically adequate is the following: when the mean value of conditional waiting time in queue does not exceed the mean service time. Examples illustrating this point are given in Table 2. The exponential distribution, with single parameter θ, which approximates the conditional c.d.f. of waiting time in queue is denoted by $\tilde{F}(t,\theta)$. The exact c.d.f. is denoted $F(t)$.

$$\tilde{F}(t,\theta) = 1 - \exp(-t/\theta) . \qquad (64)$$

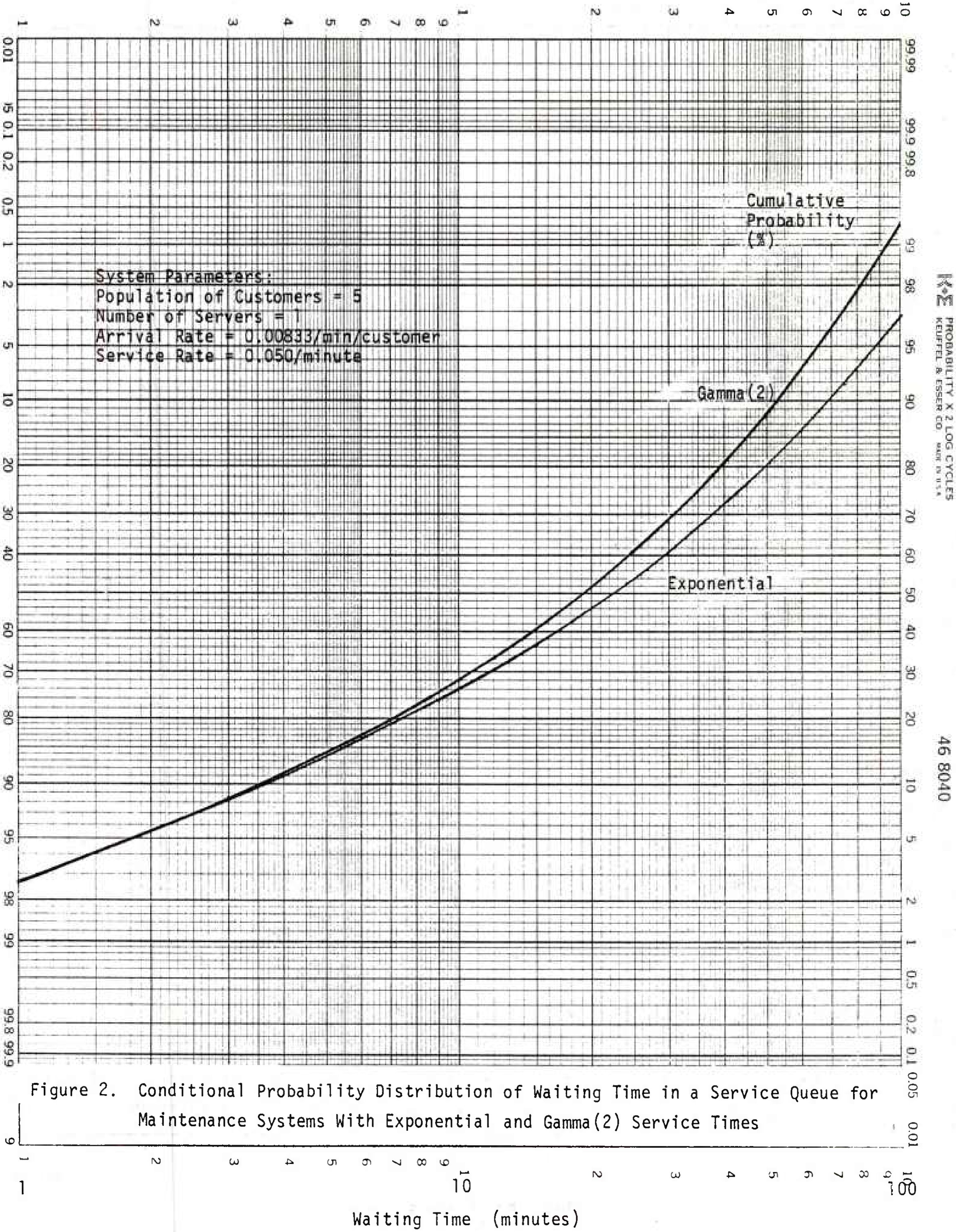


Figure 2. Conditional Probability Distribution of Waiting Time in a Service Queue for Maintenance Systems With Exponential and Gamma(2) Service Times

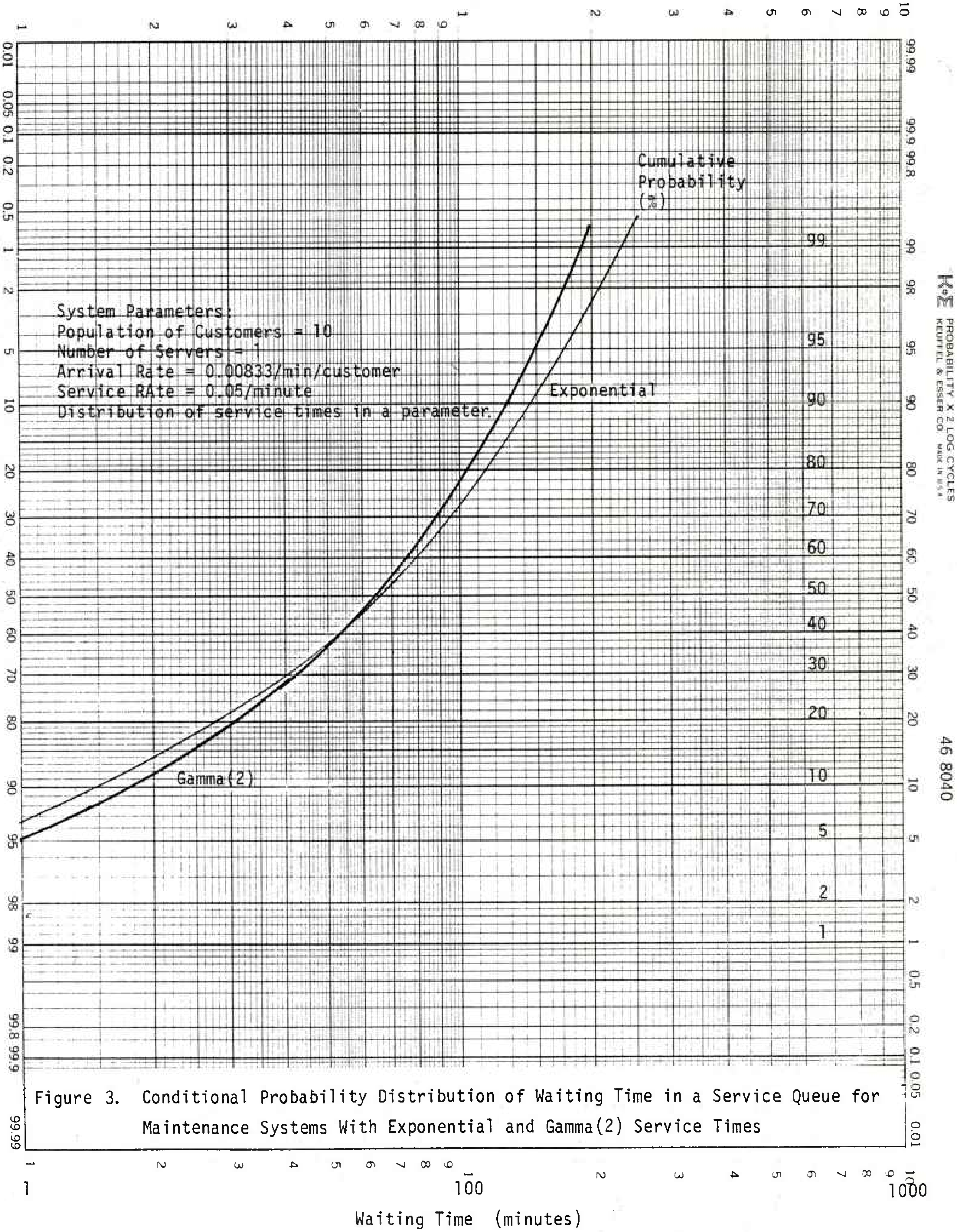


Figure 3. Conditional Probability Distribution of Waiting Time in a Service Queue for Maintenance Systems With Exponential and Gamma(2) Service Times

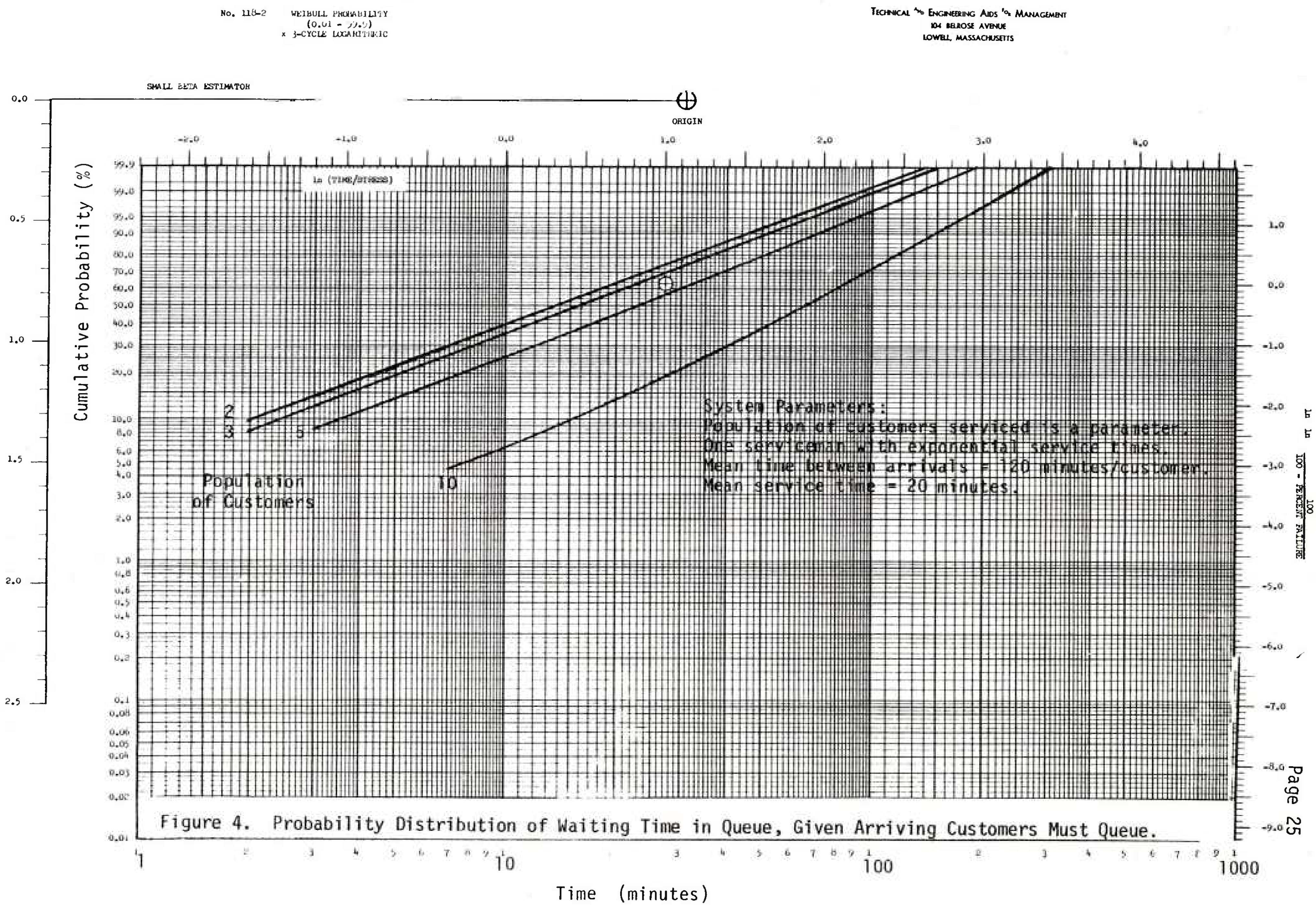


Figure 4. Probability Distribution of Waiting Time in Queue, Given Arriving Customers Must Queue.

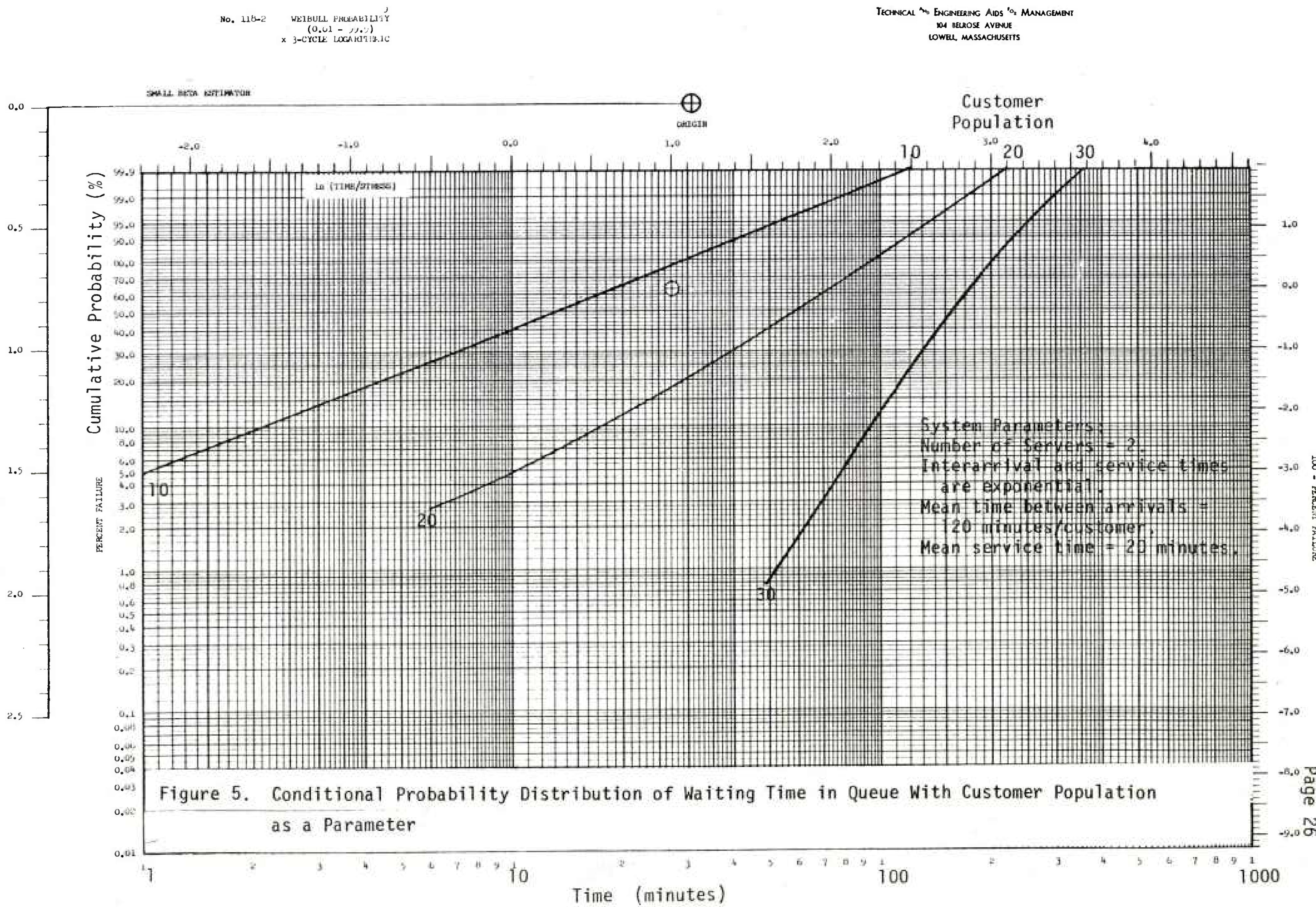


Figure 5. Conditional Probability Distribution of Waiting Time in Queue With Customer Population as a Parameter

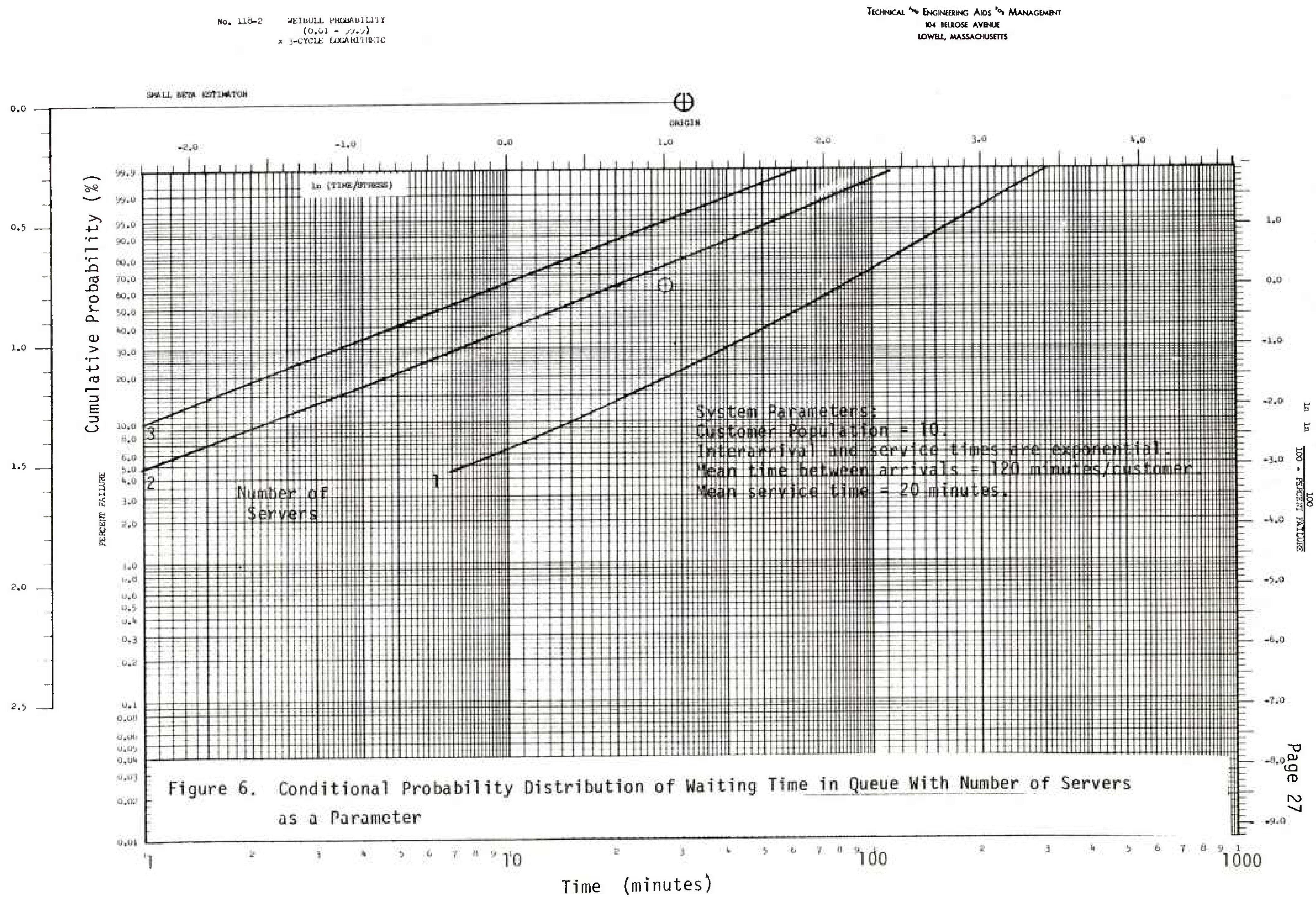


Figure 6. Conditional Probability Distribution of Waiting Time in Queue With Number of Servers as a Parameter

TABLE 2·

COMPARISON OF EXACT AND APPROXIMATE* CONDITIONAL PROBABILITY DISTRIBUTIONS OF WAITING TIME IN QUEUE FOR A SERVICE SYSTEM+ WITH SEVERAL SERVERS AND WITH A FINITE CUSTOMER POPULATION

Time, t (minutes)		Number of Servers							
		1 (left t-scale)		2 (right t-scale)		3 (right t-scale)		4 (right t-scale)	
		$F(t)$	$\tilde{F}(t, 74.6)$	$F(t)$	$\tilde{F}(t, 18.6)$	$F(t)$	$\tilde{F}(t, 9.41)$	$F(t)$	$\tilde{F}(t, 6.17)$
7	1	0.0450	0.0896	0.0478	0.0523	0.0984	0.1008	0.1482	0.1496
14	2	0.0944	0.1711	0.0936	0.1019	0.1874	0.1915	0.2747	0.2768
21	5	0.1472	0.2454	0.2199	0.2357	0.4063	0.4122	0.5530	0.5553
28	10	0.2025	0.3129	0.3964	0.4159	0.6505	0.6545	0.8017	0.8022
56	20	0.4308	0.5279	0.6467	0.6588	0.8815	0.8806	0.9617	0.9609
98	30	0.7156	0.7312	0.7989	0.8007	0.9609	0.9587	0.9928	0.9923
140	40	0.8823	0.8469	0.8883	0.8836	0.9874	0.9857	0.9987	0.9985
196	50	0.9711	0.9277	0.9393	0.9320	0.9960	0.9951	0.9998	0.9997
280	60	0.9975	0.9766	0.9677	0.9603	0.9988	0.9983	1.0000	1.0000

* The approximation is an exponential distribution of the form $\tilde{F}(t,\theta) = 1 - \exp(-t/\theta)$. The exact distribution is denoted by $F(t)$.

\+ Each customer has an exponentially distributed interarrival time. Service times are exponential.
Mean time between interarrivals = 120 minutes/customer.
Customer population = 10.
Mean service time = 20 minutes.

The parameter θ is assigned the exact conditional mean waiting time ($W_{q|w}$). This value can be obtained from easily computed quantities as follows. The system mean waiting time is calculated from (21). Then, W_q is calculated via (20). Both of these relations exploit Little's formula. Then, Bayes' theorem requires

$$W_{q|w} = W_q/(1-p_q) \ , \tag{65}$$

where p_q is the probability that an arrival must queue. With c servers,

$$p_q = 1 - \Sigma_{n=0}^{c-1} q_n \ . \tag{66}$$

For a single server,

$$q_0 = 1 - p_q \tag{67}$$

$$q_0 = mp_0/\Sigma_{n=1}^{m-1} (m-n) \ p_n \ . \tag{68}$$

Four pairs of c.d.f.'s are shown in Table 2. The parameter which distinguishes each pair is the number of servers, which varies from 1 to 4. The constants across comparisons are customer population size, mean interarrival time per customer, and mean service time. The mean service time is 20 minutes. Note that the conditional mean waiting time is less than 20 minutes in the last three comparisons, but not the first. In the first case, with a single server the approximation is poor; whereas, in the other cases the approximation is quite good.

19. Conclusions

The inferences drawn from the numerical examples are summarized here.

- o Unless the activity parameter ($m\lambda/\mu$) is smaller than about 1/3, use of queueing theory for an infinite customer population to describe the behavior of a finite queueing model is quite inexact. Further, there are no compelling numerical reasons for using results applicable to infinite systems.

- o Numerical problems in calculating the system steady-state probability vector were non-existent using double-precision arithmetic on the PRIME 750 minicomputer.

- o If expected values of service system properties such as number of customers in the system and/or in the queue are of primary concern, the differences between finite models having exponential and $\Gamma(2)$ service times seem insignificant in practice.

- One must qualify the previous conclusion. The form of the probability distribution of service time should be selected carefully when the conditional mean time in queue must be accurate. Even with the same mean, different service time c.d.f.'s may yield quite different mean waiting times. This observation holds for exponential versus $\Gamma(2)$ service-time distributions.

- The functional form of the conditional probability distribution of waiting time in queue is generally quite complex. But, a simple approximation exists under certain conditions. For a model with exponential service times, the approximation is valid when the conditional average waiting time in queue does not exceed the mean service time.

20. Recap and Survey of Other Derivations

A review at this point is appropriate. I have discussed two mathematical models used to describe the maintenance system for production machines. Because the population of machines is finite, these queueing models treat a finite customer population. In both cases machine failure is assumed to occur stochastically at a constant rate during machine operation. A failure constitutes the birth of a service demand. Similarly, the completion of service constitutes the death of the service action. With this perspective a Markov birth-death process was used to describe a particular service system probabilistically. This single-server system completes service in a time that has an Erlang(2)-- or $\Gamma(2)$--probability distribution. For contrast, a second model was described which exhibits exponentially distributed service times. The mathematical development of the latter model was nearly complete in the literature on queueing. Only the conditional c.d.f. of waiting time in queue needed to be derived for the multiple-server case. This was done next. Without further mathematical developments, I presented some numerical results. Inferences were made from these results. In the remainder of this memo I will generalize the analytic results of the finite $\Gamma(2)$ model to treat several servers. This generalization proceeds in steps. First, two servers are considered. The system state-transition diagram is constructed. From this the Kolmogorov equations are written. Then the method of numerical solution is indicated and certain statistical properties are derived. In deriving these results, equations for the general multiple server case are presented. Finally, a finite $\Gamma(2)$ model with three servers is considered. The same procedure for deriving results is followed here as was used for the one- and two-server cases.

21. Two Servers With Erlang(2) Service

The equations derived in paragraphs 6 thru 14 above apply to a service system with a single server. This case was selected initially because of its mathematical simplicity. Frequently, several repairmen work independently and in parallel on machines of (a segment of) a production system. To introduce the treatment of multiple servers, consider the case where 2 servers are present. Each is assumed to have an identical service time distribution which (again) is Erlang(2). The derivation of results follows the pattern used for a single server, with appropriate modification in the definition of system states. In labeling the states of the system, it is convenient to start with a notation which uses double subscripts. Later, the doubly subscripted state variable is replaced with a variable having a single subscript. The procedure followed in this case is easily generalized to cases having more servers. An identical approach was used in an analytic communications model for artillery with CLGP Copperhead (p. 7, [5]).

22. The state of a two-server system is characterized by the number of customers in the system and by the stage of service of each of the servers. The first server is arbitrarily considered to take the first customer. Thus, when only one customer is in the system, the first server is either in the first stage of service or the second, with the other (2nd) server idle. This situation can be represented iconically by the two system configurations:

$$\begin{matrix} 0 & 1 & 0 \\ & 0 & 0 \end{matrix} \qquad \text{or} \qquad \begin{matrix} 0 & 0 & 1 \\ & 0 & 0 \end{matrix},$$

where the first integer of the first row is the number of customers in queue and where a 1 is placed in the active stage of service for each server. States are symbolized by the label E_{ij}, where the index i equals the number of customers (k) in the system and the second index j takes on values from 1 to 3 indicating service configuration. All of the state configurations and indices are shown in Table 3.

---

[5] Schlenker, G. "Field Artillery Communication Studies with Application to CLGP," Systems Analysis Directorate Activities Summary for April 1977, May 1977.

TABLE 3

STATES OF THE SERVICE SYSTEM WITH
TWO SERVERS AND TWO STAGES OF SERVICE

Label	Configuration			Customers in
(i,j)	Q	Stage 1	Stage 2	the System
0	0	0	0	0
		0	0	
1,1	0	1	0	1
		0	0	
1,2	0	0	1	1
		0	0	
2,1	0	1	0	2
		1	0	
2,2	0	0	1	2
		1	0	
2,3	0	0	1	2
		0	1	
3,1	1	1	0	3
		1	0	
3,2	1	0	1	3
		1	0	
3,3	1	0	1	3
		0	1	
Generally, for $2 \leq k \leq m$.				
k,1	k-2	1	0	k
		1	0	
k,2	k-2	0	1	k
		1	0	
k,3	k-2	0	1	k
		0	1	

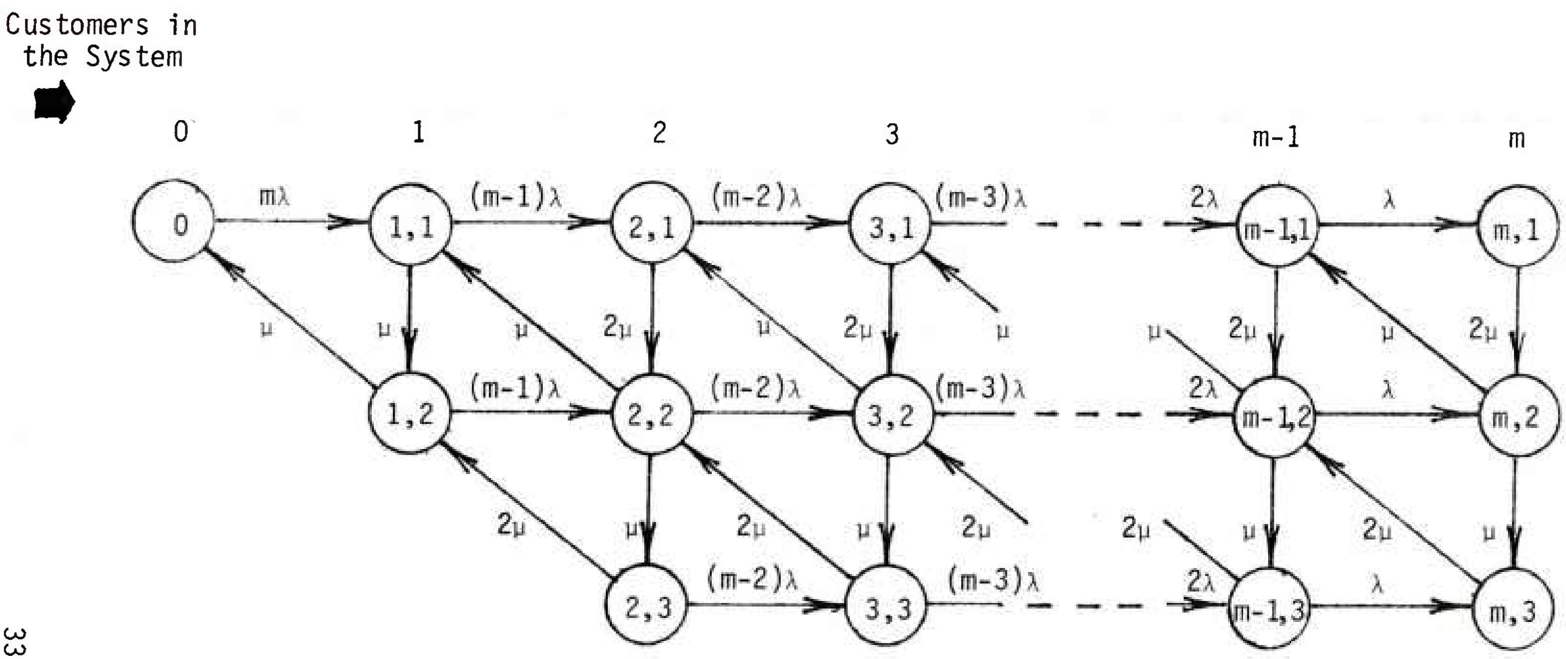


Figure 7. State Transition Diagram for a Two-Server System with Erlang (2) Service Times

23. Kolmogorov Equations for a Two-Server System

As was done for the single-server system, one can construct a state transition diagram for this system preparatory to writing the Kolmogorov equations. However, the states in this diagram are labeled with the double-index notation. This diagram is shown in Figure 7. We use the same notation for the unit birth rate (λ) into the system and the unit death rate (μ) out of the system. The definitions of λ and μ are given in equations (1) and (3). As assumed earlier, the time between arrivals for customers outside the system is exponential. Since each server is assumed to act independently, the transition rate from the $E_{1,1}$ state to the $E_{1,2}$ state is μ (the stagewise service rate for one server). If both servers worked together, the mean stage service time at this point would be less than μ^{-1}.

24. For this system define

$$P_{ij}(t) = P\{\text{system is in state } E_{ij} \text{ at time } t\} \; . \tag{69}$$

Then, using the state transition diagram, one can write the Kolmogorov equations by inspection. Notational dependence on time is suppressed. For the non-queueing states,

$$\dot{P}_0 = -m\lambda P_0 + \mu P_{12}$$

$$\dot{P}_{11} = m\lambda P_0 - ((m-1)\lambda + \mu)P_{11} + \mu P_{22}$$

$$\dot{P}_{12} = \mu P_{11} - ((m-1)\lambda + \mu)P_{12} + 2\mu P_{23}$$

$$\dot{P}_{21} = (m-1)\lambda P_{11} - ((m-2)\lambda + 2\mu)P_{21} + \mu P_{32}$$

$$\dot{P}_{22} = (m-1)\lambda P_{12} + 2\mu P_{21} - ((m-2)\lambda + 2\mu)P_{22} + 2\mu P_{33}$$

$$\dot{P}_{23} = \mu P_{22} - ((m-2)\lambda + 2\mu)P_{23} \; . \tag{70}$$

For i customers in the system with

$$3 \le i \le m-1 \; ,$$

$$\dot{P}_{i,1} = (m-i+1)\lambda P_{i-1,1} - ((m-i)\lambda + 2\mu)P_{i,1} + \mu P_{i+1,2}$$

$$\dot{P}_{i,2} = (m-1+1)\lambda P_{i-1,2} + 2\mu P_{i,1} - ((m-i)\lambda + 2\mu)P_{i,2} + 2\mu P_{i+1,3}$$

$$\dot{P}_{i,3} = (m-i+1)\lambda P_{i-1,3} + \mu P_{i,2} - ((m-i)\lambda + 2\mu)P_{i,3} \; . \tag{71}$$

The last three states are described by

$$\dot{P}_{m,1} = \lambda P_{m-1,1} - 2\mu P_{m,1}$$

$$\dot{P}_{m,2} = \lambda P_{m-1,2} + 2\mu P_{m,1} - 2\mu P_{m,2}$$

$$\dot{P}_{m,3} = \lambda P_{m-1,3} + \mu P_{m,2} - 2\mu P_{m,3} \ . \tag{72}$$

25. Steady-State Equations (Two Servers)

To obtain the steady-state solution to the above equations, all derivatives are set to zero. At this point, the double-subscript notation is changed to a single subscript notation in denoting states. Let the probability of the k th state, in stochastic steady state, be denoted by p_k, so that

$$p_0 = P_0(\infty)$$

$$p_1 = P_{11}(\infty)$$

$$p_2 = P_{12}(\infty)$$

$$p_3 = P_{21}(\infty)$$

$$p_4 = P_{22}(\infty)$$

$$p_5 = P_{23}(\infty) \tag{73a}$$

In general, for $2 \leq i \leq m$ and $1 \leq j \leq 3$,

$$k = 3(i-1) + j-1$$

and

$$p_{3i+j-4} = P_{ij}(\infty) \ . \qquad 73b)$$

Note that max(k) = 3m-1.

With this notational change, the steady-state system equations are, from (70, 71, 72, 73) ,

$$-m\lambda p_0 + \mu p_2 = 0$$

$$m\lambda p_0 - ((m-1)\lambda + \mu)p_1 + \mu p_4 = 0$$

$$\mu p_1 - ((m-1)\lambda + \mu)p_2 + 2\mu p_5 = 0$$

$$(m-1)\lambda p_1 - ((m-2)\lambda + 2\mu)p_3 + \mu p_7 = 0$$

$$(m-1)\lambda p_2 + 2\mu p_3 - ((m-2)\lambda + 2\mu)p_4 + 2\mu p_8 = 0$$

$$\mu p_4 - ((m-2)\lambda + 2\mu)p_5 = 0 \ . \tag{74}$$

Since the probabilities $\{p_k\}$ must sum to unity, the first (redundant) equation in (74) can be discarded and p_0 eliminated from the second via the substitution

$$p_0 = 1 - \Sigma_{k=1}^{3m-1} p_k \ . \tag{75}$$

This operation permits the equations for $\{p_k\}$ to be written in matrix-vector form:

$$A\ \underline{p} = \underline{b} \ , \tag{76}$$

where

$\underline{p}$ and $\underline{b}$ are $(3m-1 \times 1)$ and A is $(3m-1 \times 3m-1)$.

From the second equation in (74), with (75),

$$a_{11} = -2m\lambda + \lambda - \mu$$

$$a_{14} = \mu - m\lambda$$

$$a_{1j} = -m\lambda \quad , \quad 1 < j < 3m-1 \text{ and } j \neq 4 \ . \tag{77}$$

Also

$$\underline{b}' = (-m\lambda\ ,\ 0,\ 0,\ \ldots,\ 0) \ . \tag{78}$$

The elements of A--$\{a_{rc}\}$-- can be assigned in groups of 3 rows each as follows:

For $6 \leq k \leq 3m-4$, corresponding to $3 \leq i \leq m-1$ and $1 \leq j \leq 3$ in the double-subscript notation,

$$\text{let} \quad r = 3(i-1) \ . \tag{79a}$$

Then,

$$a_{r,3(i-2)} = (m-i+1)\lambda$$

$$a_{r,3(i-1)} = -((m-i)\lambda + 2\mu)$$

$$a_{r,3(i+1)} = \mu \ . \tag{79b}$$

Now assign

$$r = 3(i-1) + 1 \tag{80a}$$

in

$$a_{r,3(i-2)+1} = (m-i+1)\lambda$$

$$a_{r,3(i-1)} = 2\mu$$

$$a_{r,3(i-1)+1} = -((m-i)\lambda + 2\mu)$$

$$a_{r,3i+2} = 2\mu \ . \tag{80b}$$

Finally, for

$$r = 3(i-1) + 2 \ , \tag{81}$$

$$a_{r,3(i-2)+2} = (m-i+1)\lambda$$

$$a_{r,3(i-1)+1} = \mu$$

$$a_{r,3(i-1)+2} = -((m-1)\lambda + 2\mu) \ . \tag{82}$$

For the last three rows in A,

$$a_{3m-3,3m-6} = \lambda$$

$$a_{3m-3,3m-3} = -2\mu$$

$$a_{3m-2,3m-5} = \lambda$$

$$a_{3m-2,3m-3} = 2\mu$$

$$a_{3m-2,3m-2} = -2\mu$$

$$a_{3m-1,3m-4} = \lambda$$

$$a_{3m-1,3m-2} = \mu$$

$$a_{3m-1,3m-1} = -2\mu \ . \tag{83}$$

As in the case with a single server, the state probability vector ($\underline{p}$) is obtained via (14), with $\underline{b}$ and A given in (78) thru (83).

26. Statistics for the Two-Server System

To calculate the mean and variance of number of customers in the system requires aggregation of the system-state probabilities. Recall that π_k is defined as the probability that k customers are in the system. Referring to Table 3 for the definition of states,

$$\pi_0 = P_0$$

$$\pi_1 = P_{11} + P_{12}$$

$$\pi_k = \Sigma_{j=1}^{3} P_{kj} \quad , \quad 2 \le k \le m . \tag{84a}$$

Using the single-subscript notation, from (72 and 73),

$$\pi_0 = p_0$$

$$\pi_1 = p_1 + p_2$$

$$\pi_k = p_{3k-3} + p_{3k-2} + p_{3k-1} \quad , \quad 2 \le k \le m . \tag{84b}$$

The mean and variance of number of customers in the system are obtained from (16) and (17). To generalize (84b) for a system with c servers, note that

$$\pi_0 = p_0$$

$$\pi_1 = p_1 + p_2$$

$$\pi_2 = p_3 + p_4 + p_5$$

$$\vdots$$

$$\pi_{c-1} = \Sigma_{j=\ell-c+1}^{\ell} \tag{85a}$$

with

$$\ell = c(c+1)/2 - 1 . \tag{85b}$$

For $\quad c \le k \le m$,

$$\pi_k = \Sigma_{j=(c+1)k-\ell-1}^{(c+1)k-\ell-1+c} p_j . \tag{85c}$$

With multiple servers, the expected value and variance of the number of busy servers (N_{busy}) are:

$$E[N_{busy}] = \Sigma_{k=1}^{c} k\, \pi_k + c\Sigma_{k=c+1}^{m} \pi_k \tag{86}$$

and

$$V[N_{busy}] = \Sigma_{k=1}^{c} k^2 \pi_k + c^2 \Sigma_{k=c+1}^{m} \pi_k - (E[N_{busy}])^2 \,. \tag{87}$$

With c servers, a queue forms when the number of customers in the system N_{sys} exceeds c. The number in queue

$$N_q = N_{sys} - c \quad , \qquad N_{sys} \geq c \,. \tag{88}$$

From this

$$\overline{N}_q = \Sigma_{k=c+1}^{m} (k-c)\pi_k \tag{89}$$

and

$$V[N_q] = \Sigma_{k=c+1}^{m} (k-c)^2 \pi_k - \overline{N}_q^2 \,. \tag{90}$$

In all these cases Little's formula applies. Hence, the mean waiting time in the system (W) and the mean waiting time in the queue (W_q) are obtained via (21) and (22), respectively.

27. For the specific case of c equal to 2 (with which we are dealing), the conditional state probability density (q_k) is obtained from p_k via a procedure similar to that followed for the single-server case (pgf. 11). In employing Bayes' theorem, states having the same number of customers in the system are grouped. Thus,

$$q_0 = mp_0/K$$

$$q_1 = (m-1)p_1/K$$

$$q_2 = (m-1)p_2/K \;;$$

and, in general for $2 \leq k \leq m-1$ and $1 \leq j \leq 3$,

$$q_{3(k-1)+j-1} = K^{-1}\,(m-k)p_{3(k-1)+j-1} \,. \tag{91}$$

This set of equations contrasts with (24) for a single server. The constant K is obtained by summing over all states, equating the sum to unity. In the general multiple-server case, the conditional state p.d.f. is obtained from

$$q_0 = mp_0/K$$

$$q_1 = (m-1)p_1/K$$

$$q_2 = (m-1)p_2/K$$

$$q_3 = (m-2)p_3/K$$

$$q_4 = (m-2)p_4/K$$

$$q_5 = (m-2)p_5/K$$

...thru all states with <u>less</u> than c customers in the system. The last such state has index ℓ, where

$$\ell = (c-1)(c+2)/2$$

or

$$\ell = c(c+1)/2 - 1 \ . \tag{92a}$$

Thus,

$$q_\ell = (m-c+1)p_\ell/K \ . \tag{92b}$$

Additionally, for the states in which the number of customers (k) in the system $\geq$ c:

$$q_i = (m-k)p_i/K \ , \tag{93a}$$

with

$$i = (c+1)(k-c) + j + \ell \ , \tag{93b}$$

$$1 \leq j \leq c+1 \quad , \quad c \leq k \leq m-1 \ .$$

For 2 servers, the conditional probability that an arriving customer finds k others in the system is given by

$$\rho_0 = q_0$$

$$\rho_1 = q_1 + q_2$$

$$\rho_k = \Sigma_{j=1}^{3} q_{3(k-1)+j-1} \quad , \quad 2 \leq k \leq m-1 \ . \tag{94}$$

This result corresponds to equation (26) for a single-server system. A generalization of this result to a c-server system is the following:

$$\rho_0 = q_0$$

$$\rho_1 = q_1 + q_2$$

$$\rho_2 = q_3 + q_4 + q_5$$

$$\vdots$$

$$\rho_{c-1} = \Sigma_{j=1}^{c} q_{\ell-c+j} \; . \qquad (95a)$$

And,

$$\rho_k = \Sigma_{j=1}^{c+1} q_{(c+1)(k-c)+\ell+j} \quad , \quad c \le k \le m-1 \; . \qquad (95b)$$

28. Three Servers With Erlang(2) Service

Derivation of the state equations for the three-server system with Erlang(2) service times follows the pattern used for the two-server system (pgf. 21 ff). To facilitate recognition of states, double subscript notation is also used here. The state E_{ij} signifies that i customers are in the system, with the service stages for the three channels occupied indicated by subscript j. The max value of j is c+1 or 4, in this case. These states are depicted iconically and are labeled in Table 4. Note that the last state for which no queueing occurs is $E_{3,4}$. The state transition diagram for this system is shown in Figure 8. As indicated earlier, one can immediately write the Kolmogorov equations by inspection from the state-transition diagram. Parenthetically, the transient version of this model was solved for a communication system having the same transition-state diagram. See p. 14 of [5]. The Kolmogorov equations, in double-subscript notation, yield the time-dependent probabilities $P_{ij}(t)$, for admissible values of i and j. Since the steady-state solution is of interest here, time derivatives are set to zero. Further, the development of a matrix-vector equation for the steady-state probabilities requires conversion to single-subscript notation.

---

[5] Schlenker, G. Op. Cit., May 1977.

TABLE 4

STATES OF THE SERVICE SYSTEM WITH
THREE SERVERS AND TWO STAGES OF SERVICE

Label (i,j)	Configuration Q	Stage 1	Stage 2	Customers in the System
0	0	0	0	0
		0	0	
		0	0	
1,1	0	1	0	1
		0	0	
		0	0	
1,2	0	0	1	1
		0	0	
		0	0	
2,1	0	1	0	2
		1	0	
		0	0	
2,2	0	0	1	2
		1	0	
		0	0	
2,3	0	0	1	2
		0	1	
		0	0	
3,1	0	1	0	3
		1	0	
		1	0	
3,2	0	0	1	3
		1	0	
		1	0	
3,3	0	0	1	3
		0	1	
		1	0	
3,4	0	0	1	3
		0	1	
		0	1	

TABLE 4 (Cont)

Label (i,j)	Q	Configuration Stage 1	Stage 2	Customers in the System
4,1	1	1	0	4
		1	0	
		1	0	
4,2	1	0	1	4
		1	0	
		1	0	
4,3	1	0	1	4
		0	1	
		1	0	
4,4	1	0	1	4
		0	1	
		0	1	
For k customers in the system, $3 \leq k \leq m$.				
k,1	k-3	1	0	k
		1	0	
		1	0	
k,2	k-3	0	1	k
		1	0	
		1	0	
k,3	k-3	0	1	k
		0	1	
		1	0	
k,4	k-3	0	1	k
		0	1	
		0	1	

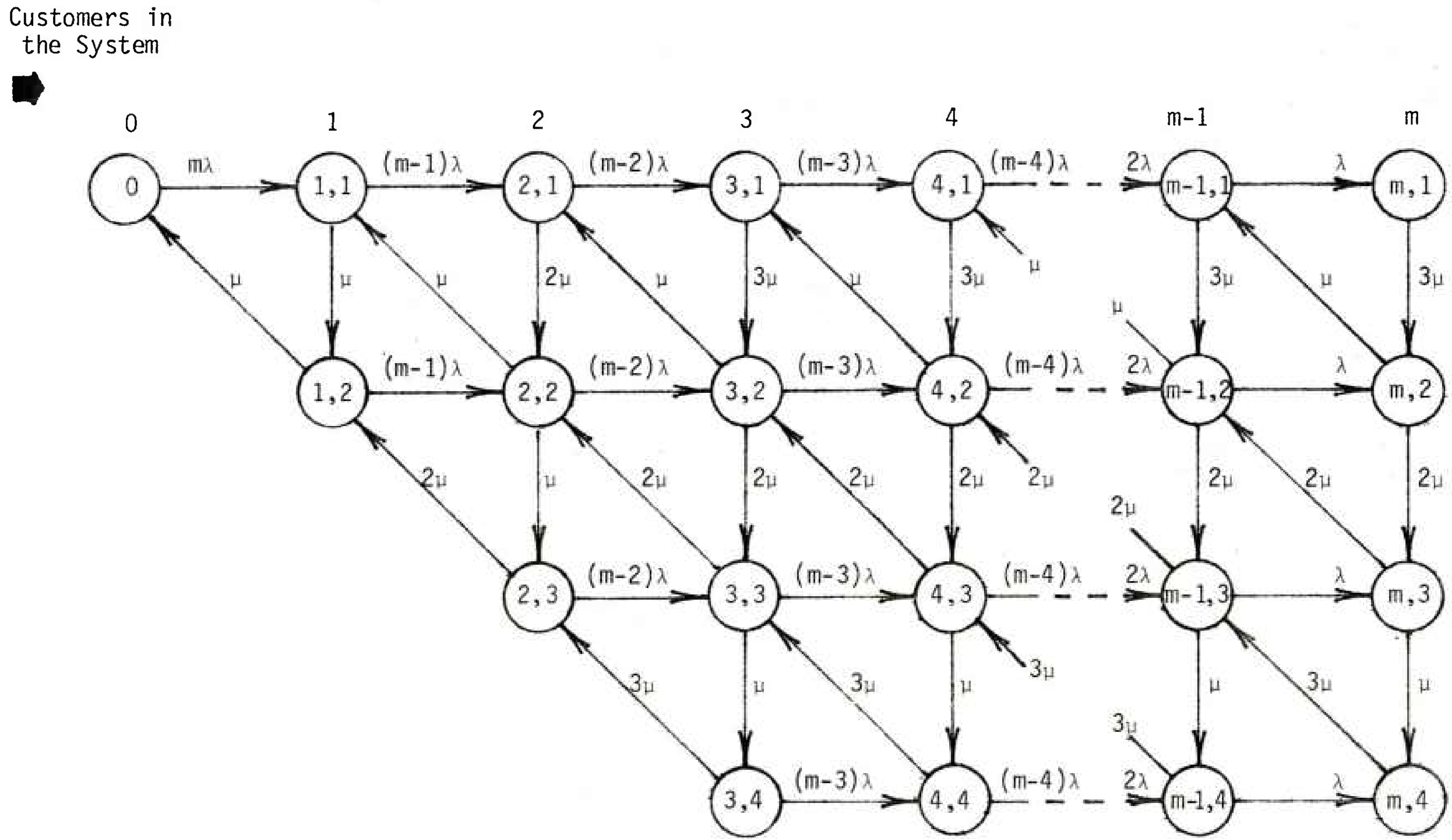


Figure 8. State Transition Diagram for a Three-Server System with Erlang (2) Service Times

29. Steady-State Equations (Three Servers)

The transformation from double to single subscript notation in (73a) is also used for this case. Additionally,

$$p_6 = P_{31}(\infty)$$

$$p_7 = P_{32}(\infty)$$

$$p_8 = P_{33}(\infty)$$

$$p_9 = P_{34}(\infty) \quad .$$

In general, for $3 \leq n \leq m$,

$$p_{4n-6} = P_{n,1}(\infty)$$

$$p_{4n-5} = P_{n,2}(\infty)$$

$$p_{4n-4} = P_{n,3}(\infty)$$

$$p_{4n-3} = P_{n,4}(\infty) \quad . \tag{96}$$

With these substitutions, the Kolmogorov equations in [5] become:

$$-m\lambda p_0 + \mu p_2 = 0$$

$$m\lambda p_0 - [(m-1)\lambda + \mu]p_1 + \mu p_4 = 0$$

$$\mu p_1 - [(m-1)\lambda + \mu]p_2 + 2\mu p_5 = 0$$

$$(m-1)\lambda p_1 - [(m-2)\lambda + 2\mu]p_3 + \mu p_7 = 0$$

$$(m-1)\lambda p_2 + 2\mu p_3 - [(m-2)\lambda + 2\mu]p_4 + 2\mu p_8 = 0$$

$$\mu p_4 - [(m-2)\lambda + 2\mu]p_5 + 3\mu p_9 = 0$$

$$(m-2)\lambda p_3 - [(m-3)\lambda + 3\mu]p_6 + \mu p_{11} = 0$$

$$(m-2)\lambda p_4 + 3\mu p_6 - [(m-3)\lambda + 3\mu]p_7 + 2\mu p_{12} = 0$$

$$(m-2)\lambda p_5 + 2\mu p_7 - [(m-3)\lambda + 3\mu]p_8 + 3\mu p_{13} = 0$$

$$\mu p_8 - [(m-3)\lambda + 3\mu]p_9 = 0 \quad .$$

For $4 \leq n \leq m-1$,

$$(m-n+1)\lambda p_{4n-10} - [(m-n)\lambda + 3\mu]p_{4n-6} + \mu p_{4n-1} = 0$$

$$(m-n+1)\lambda p_{4n-9} + 3\mu p_{4n-6} - [(m-n)\lambda + 3\mu]p_{4n-5} + 2\mu p_{4n} = 0$$

$$(m-n+1)\lambda p_{4n-8} + 2\mu p_{4n-5} - [(m-n)\lambda + 3\mu]p_{4n-4} + 3\mu p_{4n+1} = 0$$

$$(m-n+1)\lambda p_{4n-7} + \mu p_{4n-4} - [(m-n)\lambda + 3\mu]p_{4n-3} = 0 \quad .$$

The last four states of the system are described by:

$$\lambda p_{4m-10} - 3\mu p_{4m-6} = 0$$

$$\lambda p_{4m-9} + 3\mu p_{4m-6} - 3\mu p_{4m-5} = 0$$

$$\lambda p_{4m-8} + 2\mu p_{4m-5} - 3\mu p_{4m-4} = 0$$

$$\lambda p_{4m-7} + \mu p_{4m-4} - 3\mu p_{4m-3} = 0 \quad . \tag{97}$$

This set of equations contains a redundancy since the state probabilities must sum to unity. To eliminate this redundancy one can substitute

$$p_0 = 1 - \Sigma_{k=1}^{4m-3} p_k \tag{98}$$

into the first equation in set (97), yielding

$$m\lambda\Sigma_{k=1}^{4m-3} p_k + \mu p_2 = m\lambda \quad . \tag{99}$$

The second equation in (97), which also involves p_0, is deleted from the equations to be solved. The final solution set has the following matrix-vector form:

$$A\underline{p} = \underline{b} \ , \tag{100a}$$

where $\underline{p}$ and $\underline{b}$ are (4m-3 X 1) and A is (4m-3 X 4m-3) .
In this case,

$$\underline{b}' = (m\lambda, 0, 0, \ldots, 0) \ . \tag{100b}$$

These equations are analogous to (76), (77), and (78) in the two-server case. In the general c-server case, the dimension of $\underline{p}$ is

$$(c+1)m - (c-1)c/2 \ .$$

30. Statistics for the Three-Server System

Recall that π_k is defined as the probability that k customers are in the system. Elements of the system state probability vector $\underline{p}$ determine the elements of $\underline{\pi}$. The general relationship for a system with c servers is given in (85). Particularizing this for 3 servers,

$$\pi_0 = p_0$$

$$\pi_1 = p_1 + p_2$$

$$\pi_2 = p_2 + p_4 + p_5$$

and, for $3 \leq k \leq m$,

$$\pi_k = \Sigma_{j=4k-6}^{4k-3} p_k \ . \qquad (101)$$

Other statistics for this case are obtained from their c-server generalizations. Specifically, the mean and variance of busy servers are obtained from (86) and (87). The mean and variance of number of customers in the queue derives from (89) and (90), with mean and variance of number in the system given by (16) and (17). The conditional probability distribution for an arrival finding k other customers in the system is obtained from (95). Since Little's formula holds for a c-server system, the mean waiting time in the system (W) and in the queue (W_q) are obtained from (21) and (22), respectively. For a system with c servers, the probability that an arriving customer must wait in queue is

$$P\{\text{arrival must wait}\} = \Sigma_{k=c}^{m-1} \rho_k$$

or

$$= 1 - \Sigma_{k=0}^{c-1} \rho_k \ . \qquad (102)$$

Each of the above statistics is calculated for c = 1, 2, or 3 servers in the computer program FINITE.ME2.Q, listed in Annex A.

REFERENCES

1. DD1498, HQ US ARRCOM, DRSAR-SA, March 1983, title: Manufacturing Productivity Study.

2. Gross, Donald and Harris, Carl. Fundamentals of Queueing Theory, John Wiley and Sons, c. 1974.

3. Molina, E.C. Poisson's Exponential Binomial Limit, VanNostrand, c. 1942.

4. Schlenker, G. Numerical Methods in Renewal Theory, (AD 828276), HQ USA WECOM, February 1968.

5. Schlenker, G. "Field Artillery Communication Studies with Application to CLGP," Systems Analysis Directorate Activities Summary for April 1977, DRSAR/SA/N-66, (AD A042792), US ARRCOM, May 1977.

DISTRIBUTION LIST

No. of Copies	
1	HQDA (DAMO-ZA) Wash DC
	Commander USA DARCOM 5001 Eisenhower Ave. Alexandria, VA 22333
1	ATTN: DRCRE-IP
1	DRCPA-S
	Director, US AMSAA Aberdeen PG, MD 21005
1	ATTN: DRXSY-DL
1	DRXSY-R
	Commander, US Army Logistics Center Fort Lee, VA 23801
1	ATTN: ATCL-S
	Director, US Army Ammo Center Savanna, IL 61074
1	ATTN: SARAC-DO
12	Defense Technical Information Center Cameron Station Alexandria, VA 22314
	Commander US AMCCOM (D) Dover, NJ 07801
1	ATTN: DRSMC-QAS (D)
1	DRSMC-LCU-M (D)
1	DRSMC-LCM-E (D)
1	DRSMC-LCE-D (D)
1	DRSMC-RAA (D)
1	DRSMC-LCA-P (D)
	Commander US AMCCOM (A) Aberdeen PG, MD 21005
1	ATTN: DRSMC-CLY (A)
	Commander USA Communications and Electronics Command Ft. Monmouth, NJ 07703
1	ATTN: DRSEL-PL-SA
	Commander USA MICOM Redstone Arsenal, AL 35809
1	ATTN: DRSMI-DS

DISTRIBUTION LIST (Cont'd)

No. of Copies	
	Commander USA TACOM
	Warren, MI 48090
1	ATTN: DRSTA-V
	Commander USA TECOM
	Aberdeen PG, MD 21005
1	ATTN: DRSTE-SY
	Commander
	USA Production Base Modernization Agency
	Dover, NJ 07801
1	ATTN: Mr. S. Karlin
	Director US AMETA
	Rock Island, IL 61299
1	ATTN: DRXOM-QA
	Director USA IBEA
	Rock Island, IL 61299
1	ATTN: DRXIM-MT/Mr. S. McGlone
	Director
	US Army Inventory Research Office
	Room 800, Custom House
	2nd and Chestnut Streets
	Philadelphia, PA 19160
1	ATTN: DRXMC-IRO
	Commander US AMCCOM (R)
	Rock Island, IL 61299
1	ATTN: DRSMC-LC (R)/Mr. L. Ambrosini
1	DRSMC-IRD (R)/Mr. G. Cowan
1	DRSMC-IRI (R)/Mr. J. Kilar
1	DRSMC-PP (R)/Mr. M. Craighead
1	DRSMC-SA (R)/Mr. B. Witherspoon
1	DRSMC-SAS (R)/Mr. D. Eckman
1	Mr. F. Northey
1	Mr. J. Wasson
1	DRSMC-SAL (R)/Mr. N. Trier
1	DRSMC-SAA (R)/Mr. B. White
1	DRSMC-OPP (R)/Mr. P. Woodhouse
1	DRSMC-OPI (R)/Mr. R. Heider
2	DRSMC-LEP-L (R)/Ms. Sandi Paul
	Chief
	Benet Weapons Lab, LCWCL
	Armament R&D Center
	US AMCCOM
	Watervliet, NY 12189
1	ATTN: DRSMC-LCB-TL

ANNEX A

COMPUTER SOURCE PROGRAMS

The computer programs in this annex are used to calculate a variety of steady-state statistical properties of queueing systems having a finite customer population and a multi-server service system. Calculations are made for each of two assumptions regarding the probability distribution of service times. The random service time is treated as derived from an Erlang distribution with shape parameter 2 in the program FINITE.ME2.Q. The service time is treated as an exponential random variable in the routine BEST.SERVICE. Each of these programs call utility routines for inverting and multiplying matrices: MAT.INVERSE, MAT.MPY, and MAT.VEC.MPY. All of these programs are written in SIMSCRIPT 2.5 for the PR1ME 750 minicomputer. However, the programs do not employ features unique to this computer. Cross reference lists are included with program statements to facilitate the identification of variable type and to facilitate the location of variables in a program.

A brief (50 lines of executable code) driver program TEST.FINITE.ME2.Q reads data interactively from the terminal and echoes these inputs. This program calculates certain queueing statistics for a service system with an infinite population of customers having Poisson arrivals and with Erlang(2) service times. These quantities can be compared with statistics from systems having a finite customer population. The driver program in turn calls FINITE.ME2.Q and BEST.SERVICE. The first of these programs is largest with 415 lines of executable code. The statistics listed in the program's beginning comments are optionally printed by the routine.

The routine BEST.SERVICE has 174 lines of executable code. This program has two calculational modes: one to produce queueing statistics for the parameters supplied and another to sequentially optimize the number of servers, subject to prescribed constraints. In the present application only the calculational mode is invoked.

The major problem segments of all programs are announced via comment statements, which are identified with starting double quote marks. Inputs to the driver program are read from the terminal in response to prompting messages. Output is sent directly to the terminal for display. Since the output is lengthy, it is recommended that a COMO file be established to display or print it.

```
''TEST.FINITE.ME2.Q
''
''DRIVER PROGRAM FOR ROUTINE FINITE.ME2.Q
''SOLVES THE FINITE QUEUEING PROBLEM WITH POISSON ARRIVALS PER
''UNSERVED INDIVIDUAL IN THE POPULATION AND WITH ERLANG(2) SERVICE TIME.
''
PREAMBLE
NORMALLY MODE IS REAL
END ''OF PREAMBLE
```

Options = SEQUENCE,ID,SUBCHK,XREF,NOEXPLIST,TRACE3

```
MAIN
DEFINE N, MODE, MPOP, AND NSERVE AS INTEGER VARIABLES
DEFINE MODE.BEST AS A TEXT VARIABLE
DEFINE P.NSTATE AS A 1-DIMENSIONAL ARRAY
     LET MODE=1
     LET MODE.BEST="CALCULATE"
''
''GET INPUTS FROM THE TERMINAL.
''
'START' START NEW PAGE
     PRINT 2 LINES THUS
INPUT THE ARRIVAL RATE (PER MIN.) PER INDIVIDUAL IN THE POPULATION.
IF YOU WISH TO STOP, ENTER ZERO (0).
     READ LAMBDA
     IF LAMBDA LE 0.0
          STOP
     OTHERWISE
     PRINT 1 LINE THUS
INPUT THE SERVICE RATE (PER MINUTE).
     READ MU1
     PRINT 1 LINE THUS
INPUT THE NUMBER OF INDIVIDUALS IN THE POPULATION.
     READ MPOP
     RESERVE P.NSTATE(*) AS MPOP
     PRINT 1 LINE THUS
INPUT THE NUMBER OF SERVERS.
     READ NSERVE
     IF NSERVE GE MPOP
          PRINT 1 LINE THUS
INPUT ERROR. SERVERS > OR = CUSTOMERS.
          GO TO START
     OTHERWISE
     SKIP 4 LINES
''
''ECHO INPUT PARAMETERS.
''
     LET LAMBDA.INF=MPOP*LAMBDA
     PRINT 7 LINES WITH LAMBDA, LAMBDA.INF, MU1, MPOP, AND NSERVE
     THUS
INPUT PARAMETERS FOR A STEADY-STATE, FINITE QUEUE

LAMBDA ____   **.****** PER MINUTE PER CUSTOMER
MPOP(LAMBDA)  **.****** PER MINUTE
MU1 _______   **.****** PER MINUTE
MPOP_______   **.       CUSTOMERS
NSERVE_____   **.       SERVERS
     SKIP 2 LINES
     IF NSERVE>3
          GO TO I1
     OTHERWISE

''
''CALCULATE EXPECTED VALUES FOR SYSTEM VARIABLES FOR AN INFINITE
''POPULATION WITH ERLANG(2) SERVICE.
''
     IF LAMBDA.INF<MU1
          LET P0=1.0-LAMBDA.INF/MU1
```

```
          LET WQ=0.75*LAMBDA.INF/MU1/(MU1-LAMBDA.INF)
          LET W=WQ+1.0/MU1
          LET L=LAMBDA.INF*W ''LITTLE'S FORMULA
          LET LQ=L-1.0+P0
          SKIP 2 LINES
          PRINT 9 LINES WITH P0, WQ, W, L, LQ THUS
EXPECTED, STEADY-STATE VALUES FOR AN INFINITE POPULATION FOR ARRIVALS:
------------------------------------------------------------------------------

PROBABILITY OF AN EMPTY SYSTEM          *.****
WAITING TIME IN QUEUE  (MINUTES)      ***.****
WAITING TIME IN SYSTEM (MINUTES)      ***.****
NUMBER OCCUPYING THE SYSTEM           ***.****
NUMBER OCCUPYING THE  QUEUE           ***.****

------------------------------------------------------------------------------
          SKIP 4 LINES
      ALWAYS
 ''
 ''CALL FOR STEADY-STATE, FINITE QUEUEING CALCULATIONS.
 ''
      CALL FINITE.ME2.Q(LAMBDA,MU1,MPOP,NSERVE,MODE) YIELDING P.NULL,P.SYS.FULL,
          P.CUST.WAIT,ENSYS,SDNSYS,ENQ,ESYS.WAIT,EQ.WAIT,EQ.WAIT.GQ,SDQ.WAIT.GQ,
          E.BUSY.SERVERS, SD.BUSY.SERVERS, AND P.NSTATE(*)
 ''
 ''CALCULATE THE SYSTEM STATE FOR THE CASE WHERE SERVICE TIMES ARE EXPONENTIAL.
 ''
 'I1'CALL BEST.SERVICE(LAMBDA,MU1,MPOP,NSERVE,A,B,1,MODE.BEST) YIELDING P.NULL,
      P.SYS.FULL,P.MACH.WAIT,ENO.DOWN,SDNO.DOWN,ENO.QUEUED,ESYS.WAIT,EQ.WAIT,
      EQ.WAIT.GQ,SDQ.WAIT.GQ,E.BUSY.SERVERS,SD.BUSY.SERVERS, AND P.NSTATE(*)
      GO TO START
 END ''OF TEST.FINITE.ME2.Q
```

C R O S S - R E F E R E N C E

NAME	TYPE				MODE	LINE NUMBERS OF REFERENCES
A	RECURSIVE VARIABLE	WORD	62		DOUBLE	62
B	RECURSIVE VARIABLE	WORD	64		DOUBLE	62
BEST.SERVICE	ROUTINE				INTEGER	62
E.BUSY.SERVERS	RECURSIVE VARIABLE	WORD	58		DOUBLE	58 64
ENO.DOWN	RECURSIVE VARIABLE	WORD	68		DOUBLE	63
ENO.QUEUED	RECURSIVE VARIABLE	WORD	72		DOUBLE	63
ENG	RECURSIVE VARIABLE	WORD	48		DOUBLE	57
ENSYS	RECURSIVE VARIABLE	WORD	44		DOUBLE	57
EQ.WAIT	RECURSIVE VARIABLE	WORD	52		DOUBLE	57 63
EG.WAIT.GQ	RECURSIVE VARIABLE	WORD	54		DOUBLE	57 64
ESYS.WAIT	RECURSIVE VARIABLE	WORD	50		DOUBLE	57 63
FINITE.ME2.Q	ROUTINE				INTEGER	56
I1	LABEL					36 62
L	RECURSIVE VARIABLE	WORD	32		DOUBLE	47 48 50
LAMBDA	RECURSIVE VARIABLE	WORD	18		DOUBLE	12 13 31 32 56 62
LAMBDA.INF	RECURSIVE VARIABLE	WORD	23		DOUBLE	31 32 43 44 45* 47
LO	RECURSIVE VARIABLE	WORD	34		DOUBLE	48 50
MODE	RECURSIVE VARIABLE	WORD	2		INTEGER	2 5 56
MODE.BEST	RECURSIVE VARIABLE	WORD	5		TEXT	3 6 62
MPOP	RECURSIVE VARIABLE	WORD	3		INTEGER	2 19 20 23 31 32 56 62
MU1	RECURSIVE VARIABLE	WORD	20		DOUBLE	17 32 43 44 45* 46 56 62
N	RECURSIVE VARIABLE	WORD	1		INTEGER	2
NSERVE	RECURSIVE VARIABLE	WORD	4		INTEGER	2 22 23 32 35 56 62
P.CUST.WAIT	RECURSIVE VARIABLE	WORD	42		DOUBLE	57
P.MACH.WAIT	RECURSIVE VARIABLE	WORD	66		DOUBLE	63
P.NSTATE	RECURSIVE VARIABLE	WORD	6	(1-D)	DOUBLE	4 20* 58 64
P.NULL	RECURSIVE VARIABLE	WORD	38		DOUBLE	56 62
P.SYS.FULL	RECURSIVE VARIABLE	WORD	40		DOUBLE	56 63
PO	RECURSIVE VARIABLE	WORD	26		DOUBLE	44 48 50
SD.BUSY.SERVERS	RECURSIVE VARIABLE	WORD	60		DOUBLE	58 64
SDNO.DOWN	RECURSIVE VARIABLE	WORD	70		DOUBLE	63
SDNSYS	RECURSIVE VARIABLE	WORD	46		DOUBLE	57
SDG.WAIT.GQ	RECURSIVE VARIABLE	WORD	56		DOUBLE	57 64
START	LABEL					10 25 65
UIB.W	IMPLIED SUBSCRIPT	SYS	4		INTEGER	27 34 49 51
W	RECURSIVE VARIABLE	WORD	30		DOUBLE	46 47 50
WG	RECURSIVE VARIABLE	WORD	28		DOUBLE	45 46 50

```
ROUTINE FOR FINITE.ME2.Q (LAMBDA, MU1, MPOP, NSERVE, MODE) YIELDING P.NULL,
     P.SYS.FULL, P.CUST.WAIT, ENSYS, SDNSYS, ENQ, ESYS.WAIT, EQ.WAIT,
     EQ.WAIT.GQ, SDQ.WAIT.GQ, E.BUSY.SERVERS, SD.BUSY.SERVERS, P.NSTATE
''
''THIS ROUTINE CALCULATES THE STEADY-STATE, SYSTEM STATE-PROBABILITY
''VECTOR FOR A SERVICE SYSTEM HAVING EXPONENTIAL INTERARRIVAL TIMES
''FOR EACH MEMBER OF A FINITE CUSTOMER POPULATION, AND HAVING NSERVE SERVERS
''WITH SERVICE TIMES DISTRIBUTED AS ERLANG WITH SHAPE PARAMETER 2.
''
''INPUT:
''LAMBDA                   ARRIVAL RATE PER INDIVIDUAL IN THE POPULATION
''MU1                      SERVICE RATE OR RECIPROCAL OF MEAN SERVICE TIME
''MPOP                     CUSTOMER POPULATION SIZE
''NSERVE                   NUMBER OF SERVERS (N LE 3)
''MODE                     INTEGER FLAG FOR PRINTING FROM ROUTINE (FOR MODE=1)
''
''OUTPUT:
''P.NSTATE                 VECTOR OF PROBABILITY ELEMENTS IN WHICH THE N TH
''                         ELEMENT IS THE PROBABILITY THAT N INDIVIDUALS ARE
''                         IN THE SERVICE SYSTEM IN THE STEADY STATE.
''P.NULL                   PROBABILITY THAT THE SYSTEM IS EMPTY
''P.SYS.FULL               PROBABILITY THAT ALL SERVICE CHANNELS ARE FULL
''P.CUST.WAIT              PROBABILITY THAT AN ARRIVING CUSTOMER MUST QUEUE
''ENSYS                    EXPECTED NUMBER OF CUSTOMERS IN THE SYSTEM
''SDNSYS                   STD DEVIATION OF THE NUMBER IN THE SYSTEM
''ENQ                      EXPECTED NUMBER OF CUSTOMERS IN THE QUEUE
''ESYS.WAIT                EXPECTED WAIT IN THE SERVICE SYSTEM
''EQ.WAIT                  EXPECTED WAIT IN THE SERVICE QUEUE
''EQ.WAIT.GQ               EXPECTED QUEUE WAITING TIME, GIVEN THAT A
''                         CUSTOMER MUST QUEUE.
''SDQ.WAIT.GQ              STANDARD DEVIATION OF QUEUE WAITING TIME, GIVEN
''                         THAT A CUSTOMER MUST QUEUE.
''E.BUSY.SERVERS           EXPECTED VALUE OF BUSY SERVERS
''SD.BUSY.SERVERS          STANDARD DEVIATION OF THE NUMBER OF BUSY SERVERS
''OPTIONAL PRINTING OF THE QUEUE WAITING TIME DISTRIBUTION IS PROVIDED.
''
''ENDOGENOUS VARIABLES:
''AM                       KOLMOGOROV STATE-TRANSITION MATRIX
''AMINV                    INVERSE OF AM
''PV                       VECTOR OF PROBABILITIES IN WHICH THE N TH
''                         ELEMENT IS THE PROBABILITY OF FINDING THE
''                         SYSTEM IN THE N TH STATE, WHERE
''                         N=2*(NO CUSTOMERS-1)+STAGE OF SERVICE, NO>0.
''QV                       VECTOR OF PROBABILITIES IN WHICH THE N TH
''                         ELEMENT IS THE PROBABILITY THAT AN ARRIVING CUSTOMER
''                         FINDS THE SYSTEM IN THE N TH STATE. THIS VECTOR
''                         IS USED TO CALCULATE THE DISTRIBUTION OF
''                         WAIT IN QUEUE.
''
DEFINE I, J, MPOP, TWOM, MAX, MAXM, MODE, N, AND NSERVE AS INTEGER VARIABLES
DEFINE BV, PV, QV, P.NSTATE, AND Q.NSTATE AS REAL, 1-DIMENSIONAL ARRAYS
DEFINE AM AND AMINV AS A REAL, 2-DIMENSIONAL ARRAYS
     LET TWOM=2*MPOP
     LET MU=2.0*MU1 ''SERVICE RATE FOR EACH OF THE 2 STAGES
     LET ML=REAL.F(MPOP)*LAMBDA ''MAX ARRIVAL RATE
     RESERVE P.NSTATE(*) AS MPOP
     RESERVE Q.NSTATE(*) AS MPOP
```

```
      IF NSERVE LE 0
          RETURN
      OTHERWISE
      IF NSERVE>1
          GO TO L0
      OTHERWISE
      RESERVE BV(*) AS TWOM
      RESERVE PV(*) AS TWOM
      RESERVE GV(*) AS TWOM
      RESERVE AM(*,*) AS TWOM BY TWOM
      RESERVE AMINV(*,*) AS TWOM BY TWOM
  FOR I=1 TO TWOM DO
      LET BV(I)=0.0
      LET GV(I)=0.0
      FOR J=1 TO TWOM DO
          LET AM(I,J)=0.0
      LOOP ''OVER J
  LOOP ''OVER I TO INITIALIZE
  ''
  ''FILL NON-ZERO ELEMENTS OF THE REDUCED STATE-TRANSITION MATRIX (AM)
  ''AND THE CONSTANT VECTOR (BV).
  ''
  FOR J=I TO TWOM DO
      LET AM(I,J)=-ML
  LOOP ''OVER COLUMNS
      SUBTRACT MU+REAL.F(MPOP-1)*LAMBDA FROM AM(1,1)
      ADD MU TO AM(1,4)
      LET AM(2,1)=MU
      LET AM(2,2)=-MU-ML+LAMBDA
  FOR N=2 TO MPOP-1 DO
      LET I=2*N-1
      LET AM(I,I-2)=(MPOP-N+1)*LAMBDA
      LET AM(I,I)=-MU-(MPOP-N)*LAMBDA
      LET AM(I,I+3)=MU
      LET AM(I+1,I-1)=AM(I,I-2)
      LET AM(I+1,I)=MU
      LET AM(I+1,I+1)=AM(I,I)
  LOOP ''OVER N
  ''FINALLY,
      LET AM(TWOM-1,TWOM-3)=LAMBDA
      LET AM(TWOM-1,TWOM-1)=-MU
      LET AM(TWOM,TWOM-2)=LAMBDA
      LET AM(TWOM,TWOM-1)=MU
      LET AM(TWOM,TWOM)=-MU
      LET BV(1)=-ML
  ''
  ''OBTAIN THE INVERSE OF AM(*,*).
  ''
      CALL MAT.INVERSE(AM(*,*),TWOM) YIELDING AMINV(*,*)
  ''
  ''OBTAIN THE SOLUTION VECTOR OF THE MATRIX EQUATION.
  ''
      CALL MAT.VEC.MPY(AMINV(*,*),BV(*),TWOM) YIELDING PV(*)
  ''
  ''SOLVE FOR THE EMPTY-SYSTEM PROBABILITY (P.NULL) AND CALCULATE
  ''THE STATE-PROBABILITY VECTOR (P.NSTATE).
  ''
```

```
        LET ENSYS=0.0
        LET VNSYS=0.0
        LET ENQ=0.0
        LET ESQ=0.0
        LET SUM=0.0
    FOR N=1 TO MPOP DO
        LET I=2*N-1
        LET P.NSTATE(N)=PV(I)+PV(I+1)
        ADD P.NSTATE(N) TO SUM
        ADD N*P.NSTATE(N) TO ENSYS
        ADD N**2*P.NSTATE(N) TO VNSYS
        ADD (N-1)*P.NSTATE(N) TO ENQ
        ADD (N-1)**2*P.NSTATE(N) TO ESQ
    LOOP ''OVER NON-ZERO SYSTEM STATES
    ''
    ''CHECK VALIDITY OF PROBABILITY SUM.
    ''
        IF SUM<0.0 OR SUM>1.0
           PRINT 1 LINE WITH SUM THUS
ERROR IN ROUTINE FINITE.ME2.Q.  PARTIAL SUM OF STATE PROBABILITIES = **.***.
           STOP
        OTHERWISE
        LET P.NULL=1.0-SUM
        LET P.SYS.FULL=SUM ''FOR A SINGLE SERVER
        LET ENQ.CHECK=ENSYS-SUM ''FOR A SINGLE SERVER
    ''
    ''CALCULATE THE VARIANCE AND STD DEV OF THE NUMBER IN THE SYSTEM.
    ''
        LET VNSYS=VNSYS-ENSYS**2
        LET SDNSYS=SQRT.F(VNSYS)
        LET VNQ=ESQ-ENQ**2
        LET SDNQ=SQRT.F(VNQ)
    ''
    ''CALCULATE THE MEAN AND STANDARD DEVIATION OF BUSY SERVERS.
    ''
        LET E.BUSY.SERVERS=1.0-P.NULL ''FOR A SINGLE SERVER
        LET VAR.BUSY.SERVERS=1.0-P.NULL-E.BUSY.SERVERS**2
        LET SD.BUSY.SERVERS=SQRT.F(VAR.BUSY.SERVERS)
    ''
    ''CALCULATE MEAN WAITING TIMES USING LITTLE'S FORMULA.
    ''
        LET ESYS.WAIT=ENSYS/LAMBDA/(REAL.F(MPOP)-ENSYS)
        LET EQ.WAIT=ESYS.WAIT-1.0/MU1
        IF MPOP=1
           RETURN
        OTHERWISE ''CALCULATE AND PRINT WAITING TIME DISTRIBUTIONS
    ''
    ''CALCULATE THE PROBABILITY THAT AN ARRIVAL FINDS THE SYSTEM
    ''IN THE N TH STATE (GV(N)).
    ''
        LET NORM.CONST=MPOP*P.NULL
    FOR N=1 TO MPOP-1 DO
        ADD (MPOP-N)*P.NSTATE(N) TO NORM.CONST
    LOOP ''TO CALCULATE THE NORMALIZATION CONSTANT
        LET G.NULL=MPOP*P.NULL/NORM.CONST
        LET P.CUST.WAIT=1.0-G.NULL
        LET EQ.WAIT.GQ=0.
```

```
        LET VQ.WAIT.GQ=0.0
    FOR N=1 TO MPOP-1 DO
        LET I=2*N-1
        LET QV(I)=(MPOP-N)*PV(I)/NORM.CONST
        LET QV(I+1)=(MPOP-N)*PV(I+1)/NORM.CONST
        LET Q.NSTATE(N)=QV(I)+QV(I+1)
        ADD (I+1)*QV(I)+I*QV(I+1) TO EQ.WAIT.GQ
        ADD (I+1)*(I+2)*QV(I)+I*(I+1)*QV(I+1) TO VQ.WAIT.GQ
    LOOP ''OVER THE NUMBER OF ARRIVING CUSTOMERS
            LET EQ.WAIT.CHECK=EQ.WAIT.GQ./MU
            LET ESYS.WAIT.CHECK=EQ.WAIT.CHECK +1.0/MU1
            LET EQ.WAIT.GQ=EQ.WAIT.GQ/MU/(1.0-Q.NULL)
            LET VQ.WAIT.GQ=VQ.WAIT.GQ/MU/MU/(1.0-Q.NULL)-EQ.WAIT.GQ**2
            LET SDQ.WAIT.GQ=SQRT.F(VQ.WAIT.GQ)
            IF MODE NE 1
                RETURN
            OTHERWISE
   'L1'SKIP 4 LINES
        PRINT 21 LINES WITH NSERVE,P.SYS.FULL,ENSYS,SDNSYS,ENQ,SDNQ,E.BUSY.SERVERS,
        SD.BUSY.SERVERS,ESYS.WAIT,EQ.WAIT,EQ.WAIT.GQ,P.NULL,P.NULL,Q.NULL,Q.NULL
        THUS
STATE OF THE FINITE SYSTEM IN STEADY STATE WITH GAMMA(2) SERVICE FOR ** SERVERS

     PROBABILITY:  SYSTEM  IS FULL       *.****
     EXPECTED NUMBER IN THE SYSTEM      **.****
     STD DEV  NUMBER IN THE SYSTEM      **.****
     EXPECTED NUMBER IN THE  QUEUE      **.****
     STD DEV  NUMBER IN THE  QUEUE      **.****
     EXPECTED NUMBER  BUSY SERVERS      **.****
     STD DEV  NUMBER  BUSY SERVERS      **.****
MEAN WAITING TIME (MIN) IN SYSTEM      ***.***
MEAN WAITING TIME (MIN) IN  QUEUE      ***.***
COND'AL MEAN TIME (MIN) IN  QUEUE      ***.***

PROBABILITY DISTRIBUTIONS FOR NUMBER OF CUSTOMERS IN THE SYSTEM

----------------UNCONDITIONAL--------------GIVEN ARRIVAL-------------------------------------
NO IN            PROB      CUML            PROB      CUML
SYSTEM           DENS      PROB            DENS      PROB
---------------------------------------------------------------------------------------------
  0             *.****    *.****         *.****    *.****
        LET CUM.DIST=P.NULL
        LET CUM.DIST.COND=Q.NULL
    FOR N=1 TO MPOP DO
        ADD P.NSTATE(N) TO CUM.DIST
        ADD Q.NSTATE(N) TO CUM.DIST.COND
        PRINT 1 LINE WITH N, P.NSTATE(N), CUM.DIST, Q.NSTATE(N), CUM.DIST.COND
        THUS
 **             *.****    *.****         *.****    *.****
    LOOP ''OVER THE SYSTEM STATES
        PRINT 2 LINES THUS
---------------------------------------------------------------------------------------------

        IF NSERVE>1
            RETURN
        OTHERWISE
```

```
        LET DELT=MAX.F(1.0,TRUNC.F(EQ.WAIT.GQ/10.0))
        LET MAX=60
        SKIP 2 LINES
    ''
    ''PRINT HEADINGS FOR THE QUEUE WAITING TIME DISTRIBUTION.
    ''
        PRINT 8 LINES WITH MPOP AND Q.NULL THUS
CUMULATIVE AND CONDITIONAL CUM PROB DISTRIBUTIONS OF WAITING TIME IN QUEUE
EXPONENTIAL INTERARRIVALS FOR EACH OF ** CUSTOMERS.  ERLANG(2) SERVICE.

------------------------------------------------------------------------------
 TIME     CUML      COND
(MIN)     PROB      PROB
------------------------------------------------------------------------------
  0      *.****      0
    ''
    ''START TIME LOOP.
    ''
        FOR I=1 TO MAX ''TIME STEPS'' DO
            LET TM=I*DELT
            LET ARG=TM*MU
            LET CUM.DIST=1.0-EXP.F(-ARG)*(QV(1)*(1.0+ARG)+QV(2))
            LET SUMN=0.0
            FOR N=2 TO MPOP-1 DO
                LET SUMEJ=1.0
                LET ARGJ=1.0
                LET JFACTORIAL=1.0
                FOR J=1 TO 2*(N-1) DO
                    LET ARGJ=ARGJ*ARG
                    LET JFACTORIAL=JFACTORIAL*J
                    ADD ARGJ/JFACTURIAL TO SUMEJ
                LOOP ''OVER J
                LET ARGJ=ARGJ*ARG
                LET JFACTORIAL=JFACTORIAL*(2*N-1)
                LET SUMOJ=SUMEJ+ARGJ/JFACTORIAL
                ADD QV(2*N-1)*SUMOJ+QV(2*N)*SUMEJ TO SUMN
            LOOP ''OVER N
            LET CUM.DIST=CUM.DIST-EXP.F(-ARG)*SUMN
            LET CON.DIST=(CUM.DIST-Q.NULL)/(1.0-Q.NULL)
            PRINT 1 LINE WITH TM, CUM.DIST, AND CON.DIST THUS
***.*    *.****    *.****
        LOOP ''OVER I TIME STEPS
        PRINT 10 LINES WITH EQ.WAIT.GQ, SDQ.WAIT.GQ, EQ.WAIT.CHECK,
        ESYS.WAIT.CHECK, Q.NULL, E.BUSY.SERVERS, SD.BUSY.SERVERS
        THUS
------------------------------------------------------------------------------
MEAN    WAITING TIME IN QUEUE, GIVEN A WAIT    ***.***  MINUTES
STD DEV WAITING TIME IN QUEUE, GIVEN A WAIT    ***.***  MINUTES
MEAN    WAITING TIME IN QUEUE(UNCONDITIONAL)   ***.***  MINUTES
MEAN    WAITING TIME IN SYS  (UNCONDITIONAL)   ***.***  MINUTES
PROBABILITY THAT AN ARRIVAL FINDS SYS EMPTY     *.****
MEAN    NUMBER OF BUSY SERVERS ____________    **.****
STD DEV NUMBER OF BUSY SERVERS ____________    **.****

        RETURN
    'LO'IF NSERVE>2
```

```
            GO TO L3
        OTHERWISE
''
''RESERVE ARRAYS FOR THE CASE:  NSERVE=2
''
        LET MAXM=3*MPOP-1
        RESERVE BV(*) AS MAXM
        RESERVE PV(*) AS MAXM
        RESERVE QV(*) AS MAXM
        RESERVE AM(*,*) AS MAXM BY MAXM
        RESERVE AMINV(*,*) AS MAXM BY MAXM
''
''FILL THE REDUCED STATE-TRANSITION MATRIX (AM) AND THE CONSTANT
''VECTOR (BV).
''
FOR I=1 TO MAXM DO
        LET BV(I)=0.0
        FOR J=1 TO MAXM DO
            LET AM(I,J)=0.0
        LOOP ''OVER J
LOOP ''OVER I
FOR J=1 TO MAXM DO
        LET AM(1,J)=ML
LOOP ''OVER J
        LET AM(1,2)=AM(1,2)+MU
        LET AM(2,1)=MU
        LET AM(2,2)=-(MPOP-1)*LAMBDA-MU
        LET AM(2,5)=2.0*MU
        LET AM(3,1)=(MPOP-1)*LAMBDA
        LET AM(3,3)=-(MPOP-2)*LAMBDA-2.0*MU
        LET AM(3,7)=MU
        LET AM(4,2)=(MPOP-1)*LAMBDA
        LET AM(4,3)=2.0*MU
        LET AM(4,4)=AM(3,3)
        LET AM(4,8)=2.0*MU
        LET AM(5,4)=MU
        LET AM(5,5)=AM(4,4)
FOR N=3 TO MPOP-1 DO
        LET I=3*(N-1)
        LET AM(I,3*N-6)=(MPOP-N+1)*LAMBDA
        LET AM(I,I)=-((MPOP-N)*LAMBDA+2.0*MU)
        LET AM(I,3*N+1)=MU
        ADD 1 TO I
        LET AM(I,3*N-5)=(MPOP-N+1)*LAMBDA
        LET AM(I,3*N-3)=2.0*MU
        LET AM(I,I)=AM(I-1,I-I)
        LET AM(I,3*N+2)=2.0*MU
        ADD 1 TO I
        LET AM(I,3*N-4)=(MPOP-N+1)*LAMBDA
        LET AM(I,3*N-2)=MU
        LET AM(I,I)=-((MPOP-N)*LAMBDA+2.0*MU)
LOOP ''OVER NUMBER OF CUSTOMERS IN THE SYSTEM
        LET I=3*(MPOP-1)
        LET AM(I,3*MPOP-6)=LAMBDA
        LET AM(I,I)=-2.0*MU
        ADD 1 TO I
        LET AM(I,3*MPOP-5)=LAMBDA
```

```
      LET AM(I,3*MPOP-3)=2.0*MU
      LET AM(I,I)=-2.0*MU
      ADD 1 TO I
      LET AM(I,3*MPOP-4)=LAMBDA
      LET AM(I,3*MPOP-2)=MU
      LET AM(I,I)=-2.0*MU
      LET BV(1)=ML
  ''
  ''OBTAIN THE INVERSE OF AM(*,*).
  ''
      CALL MAT.INVERSE (AM(*,*),MAXM) YIELDING AMINV(*,*)
  ''
  ''OBTAIN THE SOLUTION VECTOR OF THE MATRIX EQUATION.
  ''
      CALL MAT.VEC.MPY (AMINV(*,*),BV(*),MAXM) YIELDING PV(*)
  ''
  ''SOLVE FOR THE EMPTY-SYSTEM PROBABILITY (P.NULL) AND CALCULATE THE
  ''CUSTOMER STATE PROBABILITY VECTOR (P.NSTATE).
  ''
      LET ENSYS=PV(1)+PV(2)
      LET P.NSTATE(1)=ENSYS
      LET VNSYS=ENSYS
      LET ENQ=0.0
      LET ESQ=0.0
      LET SUM=ENSYS
  FOR N=2 TO MPOP DO
      LET I=3*(N-1)
      LET P.NSTATE(N)=PV(I)+PV(I+1)+PV(I+2)
      ADD P.NSTATE(N) TO SUM
      ADD N*P.NSTATE(N) TO ENSYS
      ADD (N-2)*P.NSTATE(N) TO ENQ
      ADD (N-2)**2*P.NSTATE(N) TO ESQ
      ADD N**2*P.NSTATE(N) TO VNSYS
  LOOP ''OVER THE OTHER NON-ZERO SYSTEM STATES
  ''
  ''CHECK VALIDITY OF THE PROBABILITY SUM.
  ''
      IF SUM<0.0 OR SUM>1.0
            PRINT 1 LINE WITH SUM THUS
ERROR IN ROUTINE FINITE.ME2.G.  PARTIAL SUM OF STATE PROBABILITIES = **.*****.
            STOP
      OTHERWISE
      LET P.NULL=1.0-SUM
      LET P.SYS.FULL=1.0-P.NULL-P.NSTATE(1) ''FOR 2 SERVERS
  ''
  ''CALCULATE THE VARIANCE AND STANDARD DEV OF THE NUMBER IN THE SYSTEM.
  ''
      LET VNSYS=VNSYS-ENSYS**2
      LET SDNSYS=SQRT.F(VNSYS)
      LET VNQ=ESQ-ENQ**2
      LET SDNQ=SQRT.F(VNQ)
  ''
  ''CALCULATE THE MEAN AND STANDARD DEVIATION OF BUSY SERVERS.
  ''
      LET E.BUSY.SERVERS=P.NSTATE(1)+2.0*(1.0-P.NULL-P.NSTATE(1))
      LET VAR.BUSY.SERVERS=P.NSTATE(1)+4.0*(1.0-P.NULL-P.NSTATE(1))
            -E.BUSY.SERVERS**2
```

```
     LET SD.BUSY.SERVERS=SQRT.F(VAR.BUSY.SERVERS)
 ''
 ''CALCULATE MEAN WAITING TIME USING LITTLE'S FORMULA.
 ''
     LET ESYS.WAIT=ENSYS/LAMBDA/(REAL.F(MPOP)-ENSYS)
     LET EQ.WAIT=ENQ/LAMBDA/(REAL.F(MPOP)-ENSYS)
     IF MPOP LE 2
        RETURN
     OTHERWISE ''CALCULATE AND PRINT CONDITIONAL PROB DISTRIBUTIONS
 ''
 ''CALCULATE THE PROBABILITY THAT AN ARRIVAL FINDS THE SYSTEM IN STATE N.
 ''
     LET NORM.CONST=MPOP*P.NULL
 FOR N=1 TO MPOP-1 DO
     ADD (MPOP-N)*P.NSTATE(N) TO NORM.CONST
 LOOP ''TO CALCULATE THE NORMALIZATION CONSTANT
     LET Q.NULL=MPOP*P.NULL/NORM.CONST
     LET QV(1)=(MPOP-1)*PV(1)/NORM.CONST
     LET QV(2)=(MPOP-1)*PV(2)/NORM.CONST
     LET Q.NSTATE(1)=QV(1)+QV(2)
 FOR N=2 TO MPOP-1 DO
     LET I=3*(N-1)
     LET QV(I)=(MPOP-N)*PV(I)/NORM.CONST
     LET QV(I+1)=(MPOP-N)*PV(I+1)/NORM.CONST
     LET QV(I+2)=(MPOP-N)*PV(I+2)/NORM.CONST
     LET Q.NSTATE(N)=QV(I)+QV(I+1)+QV(I+2)
 LOOP ''OVER NUMBER OF ARRIVING CUSTOMERS
     LET P.CUST.WAIT=1.0-Q.NULL-Q.NSTATE(1)
 'L2'LET EQ.WAIT.GQ=EQ.WAIT/P.CUST.WAIT
     LET SDQ.WAIT.GQ=EQ.WAIT.GQ ''APPROXIMATELY
     IF MODE NE 1 RETURN
     OTHERWISE
     GO TO L1
 'L3'IF MPOP LE 3
        PRINT 1 LINE WITH MPOP AND NSERVE THUS
ERROR IN FINITE.ME2.Q.  MPOP = **.  NSERVE = **.
        STOP
     OTHERWISE
 ''
 ''RESERVE ARRAYS FOR THE CASE:  NSERVE = 3.
 ''
     LET MAXM=4*MPOP-3
     RESERVE BV(*) AS MAXM
     RESERVE PV(*) AS MAXM
     RESERVE QV(*) AS MAXM
     RESERVE AM(*,*) AS MAXM BY MAXM
     RESERVE AMINV(*,*) AS MAXM BY MAXM
 ''
 ''FILL THE REDUCED STATE TRANSITION MATRIX (AM) AND THE CONSTANT VECTOR (BV).
 ''
 FOR I=1 TO MAXM DO
     LET BV(I)=0.0
     FOR J=1 TO MAXM DO
        LET AM(I,J)=0.0
     LOOP ''OVER J
 LOOP ''OVER I
 FOR J=1 TO MAXM DO
```

```
          LET AM(1,J)=ML
      LOOP ''OVER J
          LET AM(1,2)=AM(1,2)+MU
          LET AM(2,1)=MU
          LET AM(2,2)=-((MPOP-1)*LAMBDA+MU)
          LET AM(2,5)=2.0*MU
          LET AM(3,1)=(MPOP-1)*LAMBDA
          LET AM(3,3)=-((MPOP-2)*LAMBDA+2.0*MU)
          LET AM(3,7)=MU
          LET AM(4,2)=(MPOP-1)*LAMBDA
          LET AM(4,3)=2.0*MU
          LET AM(4,4)=AM(3,3)
          LET AM(4,8)=2.0*MU
          LET AM(5,4)=MU
          LET AM(5,5)=AM(4,4)
          LET AM(5,9)=3.0*MU
          LET AM(6,3)=(MPOP-2)*LAMBDA
          LET AM(6,6)=-((MPOP-3)*LAMBDA+3.0*MU)
          LET AM(6,11)=MU
          LET AM(7,4)=AM(6,3)
          LET AM(7,6)=3.0*MU
          LET AM(7,7)=AM(6,6)
          LET AM(7,12)=2.0*MU
          LET AM(8,5)=AM(7,4)
          LET AM(8,7)=2.0*MU
          LET AM(8,8)=AM(7,7)
          LET AM(8,13)=3.0*MU
          LET AM(9,8)=MU
          LET AM(9,9)=AM(8,8)
      FOR N=4 TO MPOP-1 DO
          LET I=4*N-6 ''AS THE ROW INDEX
          LET AM(I,4*N-10)=(MPOP-N+1)*LAMBDA
          LET AM(I,I)=-((MPOP-N)*LAMBDA+3.0*MU)
          LET AM(I,4*N-1)=MU
          ADD 1 TO I ''I=4*N-5
          LET AM(I,4*N-9)=(MPOP-N+1)*LAMBDA
          LET AM(I,I-1)=3.0*MU
          LET AM(I,I)=AM(I-1,I-1)
          LET AM(I,4*N)=2.0*MU
          ADD 1 TO I ''I=4*N-4
          LET AM(I,4*N-8)=(MPOP-N+1)*LAMBDA
          LET AM(I,I-1)=2.0*MU
          LET AM(I,I)=AM(I-1,I-1)
          LET AM(I,4*N+1)=3.0*MU
          ADD 1 TO I ''I=4*N-3
          LET AM(I,4*N-7)=(MPOP-N+1)*LAMBDA
          LET AM(I,I-1)=MU
          LET AM(I,I)=AM(I-1,I-1)
      LOOP ''OVER THE NUMBER OF CUSTOMERS IN THE SYSTEM
      ''
      ''EQUATIONS FOR THE LAST FOUR STATES.
      ''
          LET I=4*MPOP-6
          LET AM(I,4*MPOP-10)=LAMBDA
          LET AM(I,I)=-3.0*MU
          ADD 1 TO I ''I=4*MPOP-5
          LET AM(I,4*MPOP-9)=LAMBDA
```

```
      LET AM(I,I-1)=3.0*MU
      LET AM(I,I)=AM(I-1,I-1)
      ADD 1 TO I ''I=4*MPOP-4
      LET AM(I,4*MPOP-8)=LAMBDA
      LET AM(I,I-I)=2.0*MU
      LET AM(I,I)=AM(I-1,I-1)
      ADD 1 TO I ''I=4*MPOP-3
      LET AM(I,4*MPOP-7)=LAMBDA
      LET AM(I,I-1)=MU
      LET AM(I,I)=AM(I-1,I-1)
      LET BV(1)=ML
  ''
  ''OBTAIN THE INVERSE OF AM(*,*).
  ''
      CALL MAT.INVERSE(AM(*,*),MAXM) YIELDING AMINV(*,*)
  ''
  ''OBTAIN THE SOLUTION VECTOR OF THE MATRIX EQUATION.
  ''
      CALL MAT.VEC.MPY(AMINV(*,*),BV(*),MAXM) YIELDING PV(*)
  ''
  ''SOLVE FOR THE EMPTY SYSTEM STATE PROBABILITY (P.NULL) AND CALCULATE
  ''THE CUSTOMER STATE PROBABILITY VECTOR (P.NSTATE(*)).
  ''
      LET P.NSTATE(1)=PV(1)+PV(2) ''FOR I CUSTOMER
      LET P.NSTATE(2)=PV(3)+PV(4)+PV(5)
      LET ENSYS=P.NSTATE(1)+2.0*P.NSTATE(2)
      LET VNSYS=P.NSTATE(1)+4.0*P.NSTATE(2)
      LET ENQ=0.0
      LET ESQ=0.0
      LET SUM=P.NSTATE(1)+P.NSTATE(2)
  FOR N=3 TO MPOP DO
      LET I=4*N-6
      LET P.NSTATE(N)=PV(I)+PV(I+1)+PV(I+2)+PV(I+3)
      ADD P.NSTATE(N) TO SUM
      ADD N*P.NSTATE(N) TO ENSYS
      ADD (N-3)*P.NSTATE(N) TO ENQ
      ADD (N-3)**2*P.NSTATE(N) TO ESQ
      ADD N**2*P.NSTATE(N) TO VNSYS
  LOOP ''OVER THE OTHER NON-ZERO STATES
  ''
  ''CHECK VALIDITY OF THE PROBABILITY SUM.
  ''
      IF SUM<0.0 OR SUM>1.0
         PRINT I LINE WITH SUM THUS
ERROR IN ROUTINE FINITE.ME2.G.  PARTIAL SUM OF STATE PROBABILITIES = **.*****.
         STOP
      OTHERWISE
      LET P.NULL=1.0-SUM
      LET P.SYS.FULL=1.0-P.NULL-P.NSTATE(1)-P.NSTATE(2) ''FOR 3 SERVERS
  ''
  ''CALCULATE THE VARIANCE AND STANDARD DEVIATION OF NUMBER IN THE SYSTEM.
  ''
      LET VNSYS=VNSYS-ENSYS**2
      LET SDNSYS=SQRT.F(VNSYS)
      LET VNQ=ESQ-ENQ**2
      LET SDNQ=SQRT.F(VNQ)
  ''
```

```
''CALCULATE THE MEAN AND STANDARD DEVIATION OF BUSY SERVERS.
''
    LET E.BUSY.SERVERS=P.NSTATE(1)+2.0*P.NSTATE(2)+3.0*(1.0-P.NULL
        -P.NSTATE(1)-P.NSTATE(2))
    LET VAR.BUSY.SERVERS=P.NSTATE(1)+4.0*P.NSTATE(2)+9.0*(1.0-P.NULL
        -P.NSTATE(1)-P.NSTATE(2))-E.BUSY.SERVERS**2
    LET SD.BUSY.SERVERS=SQRT.F(VAR.BUSY.SERVERS)
''
''CALCULATE MEAN WAITING TIME USING LITTLE'S FORMULA.
''
    LET ESYS.WAIT=ENSYS/LAMBDA/(REAL.F(MPOP)-ENSYS)
    LET EQ.WAIT=ENQ/LAMBDA/(REAL.F(MPOP)-ENSYS)
    IF MPOP=3
        RETURN
    OTHERWISE ''CALCULATE AND PRINT CONDITIONAL PROB DISTRIBUTIONS
''
''CALCULATE THE PROBABILITY THAT AN ARRIVAL FINDS THE SYSTEM IN STATE N.
''
    LET NORM.CONST=MPOP*P.NULL
FOR N=1 TO MPOP-1 DO
    ADD (MPOP-N)*P.NSTATE(N) TO NORM.CONST
LOOP ''TO CALCULATE THE NORMALIZATION CONSTANT
    LET Q.NULL=MPOP*P.NULL/NORM.CONST
    LET QV(1)=(MPOP-1)*PV(1)/NORM.CONST
    LET QV(2)=(MPOP-1)*PV(2)/NORM.CONST
    LET QV(3)=(MPOP-2)*PV(3)/NORM.CONST
    LET QV(4)=(MPOP-2)*PV(4)/NORM.CONST
    LET QV(5)=(MPOP-2)*PV(5)/NORM.CONST
    LET Q.NSTATE(1)=QV(1)+QV(2)
    LET Q.NSTATE(2)=QV(3)+QV(4)+QV(5)
FOR N=3 TO MPOP-1 DO
    LET I=4*N-6
    LET QV(I)=(MPOP-N)*PV(I)/NORM.CONST
    LET QV(I+1)=(MPOP-N)*PV(I+1)/NORM.CONST
    LET QV(I+2)=(MPOP-N)*PV(I+2)/NORM.CONST
    LET QV(I+3)=(MPOP-N)*PV(I+3)/NORM.CONST
    LET Q.NSTATE(N)=QV(I)+QV(I+1)+QV(I+2)+QV(I+3)
LOOP ''OVER NUMBER OF ARRIVING CUSTOMERS
    LET P.CUST.WAIT=1.0-Q.NULL-Q.NSTATE(1)-Q.NSTATE(2)
    GO TO L2
    END ''OF ROUTINE FINITE.ME2.Q
```

C R O S S - R E F E R E N C E

```
NAME               TYPE                                      MODE      LINE NUMBERS OF REFERENCES
AM                 RECURSIVE VARIABLE   WORD  11  (2-D)  DOUBLE     52   67*   73   81   83*  84*  85
                                                                    86   89    90   91   92*  93   94*
                                                                    97   98    99  100  101  106  251*
                                                                   260  264   266* 267  268  269  270
                                                                   271  272   273  274  275* 276  277
                                                                   278* 281   282  283  285  286  287*
                                                                   288  290   291  292  295  296  298
                                                                   299  300   302  303  304  309  399*
                                                                   407  411   413* 414  415  416  417
                                                                   418  419   420  421  422* 423  424
                                                                   425* 426   427  428  429  430* 431
                                                                   432* 433   434* 435  436* 437  438
                                                                   439* 442   443  444  446  447  448*
                                                                   449  451   452  453* 454  456  457
                                                                   458* 464   465  467  468  469* 471
                                                                   472  473*  475  476  477* 482
AMINV              RECURSIVE VARIABLE   WORD  12  (2-D)  DOUBLE     52   68*  106  110  252* 309  313
                                                                   400* 482   486
ARG                RECURSIVE VARIABLE   WORD  62         DOUBLE    216  217*  224  228  233
ARGJ               RECURSIVE VARIABLE   WORD  68         DOUBLE    221  224*  226  228* 230
BV                 RECURSIVE VARIABLE   WORD   7  (1-D)  DOUBLE     51   64*   70  102  110  248* 258
                                                                   305  313   396* 405  478  486
CON.DIST           RECURSIVE VARIABLE   WORD  74         DOUBLE    234  235
CUM.DIST           RECURSIVE VARIABLE   WORD  53         DOUBLE    192  195*  197  217  233* 234  235
CUM.DIST.COND      RECURSIVE VARIABLE   WORD  55         DOUBLE    193  196*  197
DELT               RECURSIVE VARIABLE   WORD  57         DOUBLE    204  215
E.BUSY.SERVERS     ARGUMENT             NO.   16         DOUBLE      3  149   150  189  237  352  354
                                                                   526  529
ENG                ARGUMENT             NO.   11         DOUBLE      2  117   126* 144  189  321  329*
                                                                   347  360   495  503* 521  535
ENG.CHECK          RECURSIVE VARIABLE   WORD  34         DOUBLE    138
ENSYS              ARGUMENT             NO.    9         DOUBLE      2  115   124* 138  142  155* 189
                                                                   318  319   320  323  328* 345  359*
                                                                   360  493   502* 519  534* 535
EQ.WAIT            ARGUMENT             NO.   13         DOUBLE      2  156   189  360  383  535
EQ.WAIT.CHECK      RECURSIVE VARIABLE   WORD  48         DOUBLE    180  181   237
EQ.WAIT.GQ         ARGUMENT             NO.   14         DOUBLE      3  170   177* 180  182* 183  189
                                                                   204  237   383  384
ESQ                RECURSIVE VARIABLE   WORD  19         DOUBLE    118  127*  144  322  330* 347  496
                                                                   504* 521
ESYS.WAIT          ARGUMENT             NO.   12         DOUBLE      2  155   156  189  359  534
ESYS.WAIT.CHECK    RECURSIVE VARIABLE   WORD  50         DOUBLE    181  237
EXP.F              ROUTINE                               DOUBLE    217  233
FINITE.ME2.G       ROUTINE                               INTEGER     1
I                  RECURSIVE VARIABLE   WORD   1         INTEGER    50   69*   70   71   73   88   89*
                                                                    90*  91*   92*  93*  94* 121  122*
                                                                   173  174*  175* 176* 177* 178* 214*
                                                                   215  257*  258  260  280  281  282*
                                                                   283  284*  285  286  287* 288  289*
                                                                   290  291   292* 294  295  296* 297*
                                                                   298  299   300* 301* 302  303  304*
                                                                   325  326*  376  377* 378* 379* 380*
                                                                   404* 405   407  441  442  443* 444
```

```
                                                                     445*  446   447*  448*  449   450*  451
                                                                     452*  453*  454   455*  456   457*  458*
                                                                     463   464   465*  466*  467   468*  469*
                                                                     470*  471   472*  473*  474*  475   476*
                                                                     477*  499   500*  555   556*  557*  558*
                                                                     559*  560*
J             RECURSIVE VARIABLE   WORD    2      INTEGER            50    72*   73    80*   81    223*  225
                                                                     259*  260   263*  264   406*  407   410*
                                                                     411
JFACTORIAL    RECURSIVE VARIABLE   WORD   70      DOUBLE             222   225*  226   229*  230
L0            LABEL                                                  62    241
L1            LABEL                                                  188   387
L2            LABEL                                                  383   563
L3            LABEL                                                  242   388
LAMBDA        ARGUMENT             NO.     1      DOUBLE             1     55    83    86    89    90    97
                                                                     99    155   268   270   271   273   281
                                                                     282   285   290   292   295   298   302
                                                                     359   360   415   417   418   420   427
                                                                     428   442   443   446   451   456   464
                                                                     467   471   475   534   535
MAT.INVERSE   ROUTINE                             INTEGER            106   309   482
MAT.VEC.MPY   ROUTINE                             INTEGER            110   313   486
MAX           RECURSIVE VARIABLE   WORD    4      INTEGER            50    205   214
MAX.F         ROUTINE                             INTEGER            204
MAXM          RECURSIVE VARIABLE   WORD    5      INTEGER            50    247   248   249   250   251*  252*
                                                                     257   259   263   309   313   395   396
                                                                     397   398   399*  400*  404   406   410
                                                                     482   486
ML            RECURSIVE VARIABLE   WORD   15      DOUBLE             55    81    86    102   264   305   411
                                                                     478
MODE          ARGUMENT             NO.     5      INTEGER            1     50    185   385
MPOP          ARGUMENT             NO.     3      INTEGER            1     50    53    55    56    57    83
                                                                     87    89    90    120   155   157   164
                                                                     165   166   168   172   174   175   194
                                                                     210   219   247   268   270   271   273
                                                                     279   281   282   285   290   292   294
                                                                     295   298   299   302   303   324   359
                                                                     360   361   367   368   369   371   372
                                                                     373   375   377   378   379   388   389
                                                                     395   415   417   418   420   427   428
                                                                     440   442   443   446   451   456   463
                                                                     464   467   471   475   498   534   535
                                                                     536   542   543   544   546   547   548
                                                                     549   550   551   554   556   557   558
                                                                     559
MU            RECURSIVE VARIABLE   WORD   13      DOUBLE             54    83    84    85    86    90    91
                                                                     93    98    100   101   180   182   183*
                                                                     216   266   267   268   269   271   272
                                                                     274   276   277   282   283   286   288
                                                                     291   292   296   299   300   303   304
                                                                     413   414   415   416   418   419   421
                                                                     423   424   426   428   429   431   433
                                                                     435   437   438   443   444   447   449
                                                                     452   454   457   465   468   472   476
MU1           ARGUMENT             NO.     2      DOUBLE             1     54    156   181
N             RECURSIVE VARIABLE   WORD    6      INTEGER            50    87*   88    89    90    120*  121
                                                                     122   123   124*  125*  126*  127*  165*
```

```
                                                                   166*  172*  173   174   175   176   194*
                                                                   195   196   197*  219*  223   229   231*
                                                                   279*  280   281*  282   283   285*  286
                                                                   288   290*  291   292   292   325   326
                                                                   327   328*  329*  330*  331*  368*  369*
                                                                   375*  376   377   378   379   380   440*
                                                                   441   442*  443   444   446*  449   451*
                                                                   454   456*  498*  499   500   501   502*
                                                                   503*  504*  505*  543*  544*  554*  555
                                                                   556   557   558   559   560
NORM.CONST        RECURSIVE VARIABLE   WORD  42         DOUBLE     164   166*  168   174   175   367   369*
                                                                   371   372   373   377   378   379   542
                                                                   544*  546   547   548   549   550   551
                                                                   556   557   558   559
NSERVE            ARGUMENT             NO.    4         INTEGER      1    50    58    61   189   201   241
                                                                   389
P.CUST.WAIT       ARGUMENT             NO.    8         DOUBLE       2   169   382   383   562
P.NSTATE          ARGUMENT             NO.   18  (1-D)  DOUBLE       3    51    56*  122   123   124   125
                                                                   126   127   166   195   197   319   326
                                                                   327   328   329   330   331   341   352*
                                                                   353*  369   491   492   493*  494*  497*
                                                                   500   501   502   503   504   505   515*
                                                                   526*  527*  528*  529*  544
P.NULL            ARGUMENT             NO.    6         DOUBLE       1   136   149   150   164   168   189*
                                                                   192   340   341   352   353   367   371
                                                                   514   515   526   528   542   546
P.SYS.FULL        ARGUMENT             NO.    7         DOUBLE       2   137   189   341   515
PV                RECURSIVE VARIABLE   WORD   8  (1-D)  DOUBLE      51    65*  110   122*  174   175   249*
                                                                   313   318*  326*  372   373   377   378
                                                                   379   397*  486   491*  492*  500*  547
                                                                   548   549   550   551   556   557   558
                                                                   559
Q.NSTATE          RECURSIVE VARIABLE   WORD  10  (1-D)  DOUBLE      51    57*  176   196   197   374   380
                                                                   382   552   553   560   562*
Q.NULL            RECURSIVE VARIABLE   WORD  44         DOUBLE     168   169   182   183   189*  193   210
                                                                   234*  237   371   382   546   562
QV                RECURSIVE VARIABLE   WORD   9  (1-D)  DOUBLE      51    66*   71   174   175   176*  177*
                                                                   178*  217*  231*  250*  372   373   374*
                                                                   377   378   379   380*  398*  547   548
                                                                   549   550   551   552*  553*  556   557
                                                                   558   559   560*
REAL.F            ROUTINE                               INTEGER     55    83   155   359   360   534   535
SD.BUSY.SERVERS   ARGUMENT             NO.   17         DOUBLE       3   151   189   237   355   530
SDNQ              RECURSIVE VARIABLE   WORD  38         DOUBLE     145   189   348   522
SDNSYS            ARGUMENT             NO.   10         DOUBLE       2   143   189   346   520
SDQ.WAIT.GQ       ARGUMENT             NO.   15         DOUBLE       3   184   237   384
SQRT.F            ROUTINE                               DOUBLE     143   145   151   184   346   348   355
                                                                   520   522   530
SUM               RECURSIVE VARIABLE   WORD  21         DOUBLE     119   123*  132*  133   136   137   138
                                                                   323   327*  336*  337   340   497   501*
                                                                   510*  511   514
SUMEJ             RECURSIVE VARIABLE   WORD  66         DOUBLE     220   226*  230   231
SUMN              RECURSIVE VARIABLE   WORD  64         DOUBLE     218   231*  233
SUMOJ             RECURSIVE VARIABLE   WORD  72         DOUBLE     230   231
TM                RECURSIVE VARIABLE   WORD  60         DOUBLE     215   216   235
TRUNC.F           ROUTINE                               INTEGER    204
TWOM              RECURSIVE VARIABLE   WORD   3         INTEGER     50    53    64    65    66    67*   68*
```

```
                                                                     69     72     80     97*    98*    99*    100*
                                                                     101*   106    110
UIB.W                IMPLIED SUBSCRIPT    SYS    4      INTEGER      188    206
VAR.BUSY.SERVERS     RECURSIVE VARIABLE   WORD   40     DOUBLE       150    151    353    355    528    530
VNQ                  RECURSIVE VARIABLE   WORD   36     DOUBLE       144    145    347    348    521    522
VNSYS                RECURSIVE VARIABLE   WORD   17     DOUBLE       116    125*   142*   143    320    331*   345*
                                                                     346    494    505*   519*   520
VG.WAIT.GQ           RECURSIVE VARIABLE   WORD   46     DOUBLE       171    178*   183*   184
```

```
Options = SEQUENCE,ID,SUBCHK,XREF,NOEXPLIST,TRACE3

  1  ROUTINE FOR MAT.MPY (AM, BM, NELMTS) YIELDING CM
  2  ''
  3  ''ROUTINE TO MULTIPLY SQUARE MATRICES AM AND BM OF NELMTS BY NELMTS,
  4  ''YIELDING PRODUCT MATRIX CM.
  5  ''
  6  DEFINE I, J, K, NELMTS AS INTEGER VARIABLES
  7  DEFINE AM, BM, AND CM AS REAL, 2-DIMENSIONAL ARRAYS
  8  RESERVE AM(*,*) AS NELMTS BY NELMTS
  9  RESERVE BM(*,*) AS NELMTS BY NELMTS
 10  RESERVE CM(*,*) AS NELMTS BY NELMTS
 11  FOR I=I TO NELMTS DO
 12      FOR J=1 TO NELMTS DO
 13          LET CM(I,J)=0.0
 14          FOR K=1 TO NELMTS DO
 15              ADD AM(I,K)*BM(K,J) TO CM(I,J)
 16          LOOP ''OVER K
 17      LOOP ''OVER J
 18  LOOP ''OVER I
 19  RETURN
 20  END ''OF ROUTINE MAT.MPY
```

C R O S S - R E F E R E N C E

NAME	TYPE				MODE	LINE NUMBERS OF REFERENCES						
AM	ARGUMENT	NO.	1	(2-D)	DOUBLE	1	7	8*	15			
BM	ARGUMENT	NO.	2	(2-D)	DOUBLE	1	7	9*	15			
CM	ARGUMENT	NO.	4	(2-D)	DOUBLE	1	7	10*	13	15*		
I	RECURSIVE VARIABLE	WORD	1		INTEGER	6	11*	13	15*			
J	RECURSIVE VARIABLE	WORD	2		INTEGER	6	12*	13	15*			
K	RECURSIVE VARIABLE	WORD	3		INTEGER	6	14*	15*				
MAT.MPY	ROUTINE				INTEGER	1						
NELMTS	ARGUMENT	NO.	3		INTEGER	1	6	8*	9*	10*	11	12
						14						

```
 1  ROUTINE FOR MAT.VEC.MPY (AM, BV, NELMTS) YIELDING CV
 2  ''
 3  ''ROUTINE TO MULTIPLY THE SQUARE MATRIX AM , OF NELMTS BY NELMTS,
 4  ''BY THE VECTOR BV (NELMTS BY 1), YIELDING THE VECTOR CV (NELMTS BY 1).
 5  ''
 6  DEFINE I, J, K, NELMTS AS INTEGER VARIABLES
 7  DEFINE BV AND CV AS REAL, 1-DIMENSIONAL ARRAYS
 8  DEFINE AM AS A REAL, 2-DIMENSIONAL ARRAY
 9  RESERVE AM(*,*) AS NELMTS BY NELMTS
10  RESERVE BV(*) AS NELMTS
11  RESERVE CV(*) AS NELMTS
12  FOR I=1 TO NELMTS DO
13      LET CV(I)=0.0
14      FOR K=1 TO NELMTS DO
15          ADD AM(I,K)*BV(K) TO CV(I)
16      LOOP ''OVER K
17  LOOP ''OVER I
18  RETURN
19  END ''OF ROUTINE MAT.VEC.MPY
```

C R O S S - R E F E R E N C E

NAME	TYPE				MODE	LINE NUMBERS OF REFERENCES						
AM	ARGUMENT	NO.	1	(2-D)	DOUBLE	1	8	9*	15			
BV	ARGUMENT	NO.	2	(1-D)	DOUBLE	1	7	10*	15			
CV	ARGUMENT	NO.	4	(1-D)	DOUBLE	1	7	11*	13	15*		
I	RECURSIVE VARIABLE	WORD	1		INTEGER	6	12*	13	15*			
J	RECURSIVE VARIABLE	WORD	2		INTEGER	6						
K	RECURSIVE VARIABLE	WORD	3		INTEGER	6	14*	15*				
MAT.VEC.MPY	ROUTINE				INTEGER	1						
NELMTS	ARGUMENT	NO.	3		INTEGER	1	6	9*	10	11	12	14

```
Options = SEQUENCE,IC,SUBCHK,XREF,NOEXPLIST,TRACE3

ROUTINE FOR MAT.INVERSE (AM, N) YIELDING BM
''
''ROUTINE TO OBTAIN THE INVERSE OF THE N BY N MATRIX AM VIA THE
''COMPACT FORM OF THE GAUSS-JORDAN METHOD.  INVERSE IS RETURNED
''AS BM.  AM IS LEFT UNCHANGED.
''
DEFINE I, J, K, N AS INTEGER VARIABLES
DEFINE AM AND BM AS REAL, 2-DIMENSIONAL ARRAYS
RESERVE AM(*,*) AS N BY N
RESERVE BM(*,*) AS N BY N
''
''COPY AM INTO BM.  BM IS USED FOR GAUSSIAN REDUCTION.
''
FOR I=1 TO N DO
     FOR J=1 TO N DO
          LET BM(I,J)=AM(I,J)
     LOOP ''OVER J
LOOP ''OVER I
FOR I=1 TO N DO
     LET P=BM(I,I)
     IF P=0.0
          PRINT 2 LINES WITH I THUS
ERROR IN ROUTINE MAT.INVERSE.  THE ** TH DIAGONAL ELEMENT IS ZERO.
THE MATRIX CANNOT BE INVERTED.
          STOP
     OTHERWISE
     LET BM(I,I)=1.0
     FOR J=1 TO N DO
          LET BM(I,J)=BM(I,J)/P
     LOOP ''OVER J
     FOR J=1 TO N DO ''THE SECOND J-LOOP
          IF J=I
               GO TO EOJ ''END OF J-LOOP
          OTHERWISE
          LET P=BM(J,I)
          LET BM(J,I)=0.0
          FOR K=1 TO N DO
               SUBTRACT P*BM(I,K) FROM BM(J,K)
          LOOP ''OVER K
 'EOJ'LOOP ''OVER J
LOOP ''OVER I
RETURN
END ''OF ROUTINE MAT.INVERSE
```

C R O S S - R E F E R E N C E

NAME	TYPE				MODE	LINE NUMBERS OF REFERENCES
AM	ARGUMENT	NO.	1	(2-D)	DOUBLE	1 8 9* 16
BM	ARGUMENT	NO.	3	(2-D)	DOUBLE	1 8 10* 16 20 25 27* 33 34 36*
EOJ	LABEL					31 38
I	RECURSIVE VARIABLE	WORD	1		INTEGER	7 14* 16* 19* 20* 22 25* 27* 30 33 34 36
J	RECURSIVE VARIABLE	WORD	2		INTEGER	7 15* 16* 26* 27* 29* 30 33 34 36*
K	RECURSIVE VARIABLE	WORD	3		INTEGER	7 35* 36*
MAT.INVERSE	ROUTINE				INTEGER	1
N	ARGUMENT	NO.	2		INTEGER	1 7 9* 10* 14 15 19 26 29 35
P	RECURSIVE VARIABLE	WORD	4		DOUBLE	20 21 27 33 36

```
ROUTINE FOR BEST.SERVICE GIVEN LAMBDA, MU, MPOP, NSERVE, P10LIM, FQLIM, IPRINT,
MODE YIELDING P.NULL, P.SYS.FULL, P.MACH.WAIT, ENO.DOWN, SDNO.DOWN,
ENO.QUEUED, ESYS.WAIT, EQ.WAIT, EQ.WAIT.GQ, SDQ.WAIT.GQ, E.BUSY.SERVERS,
SD.BUSY.SERVERS, AND P.NSTATE
''
''AT THE OPTION OF THE USER, THIS ROUTINE ITERATIVELY DETERMINES THE
''MINIMUM NUMBER OF REPAIRMEN (>= NSERVE) REQUIRED TO PROVIDE A GIVEN
''QUALITY OF MAINTENANCE SERVICE.  TWO MEASURES OF QUALITY ARE USED:
''(A)THE PROBABILITY THAT THE FRACTION OF MACHINES DOWN EXCEEDS 10%,
''(B)THE EXPECTED FRACTION OF MACHINES QUEUEING FOR MAINTENANCE SERVICE.
''AT LEAST ONE OF THESE MEASURES MUST SATISFY ITS ASSOCIATED REQUIREMENT.
''THE STOCHASTIC MODEL ASSUMES EXPONENTIAL TIMES BETWEEN FAILURES AND
''EXPONENTIAL SERVICE TIMES WITH COMMON MTTR FOR ALL MACHINES.
''INPUT:
''LAMBDA           ARRIVAL RATE FOR SERVICE PER INDIVIDUAL MACHINE.
''MU               SERVICE RATE PER SERVER.
''MPOP             MACHINE POPULATION SIZE.
''NSERVE           NUMBER OF SERVICE PERSONNEL.
''P10LIM           LIMIT ON PROB THAT 10% OF MACHINES ARE DOWN.
''FQLIM            LIMIT ON FRACTION OF THE MACHINES THAT WAIT FOR SERVICE.
''IPRINT           INTEGER SWITCH TO PRINT FROM THE SUBROUTINE.  PRINTING
''                 OCCURS WHEN IPRINT=1.
''MODE             ROUTINE MODE OF OPN.  IF MODE="CALCULATE", STATISTICS
''                 ARE CALCULATED FOR THE SYSTEM HAVING NSERVE REPAIRMEN.
''                 NO OTHER SYSTEMS ARE CONSIDERED.  IF MODE="OPTIMIZE",
''                 THE ROUTINE WILL SELECT AN OPTIMUM NUMBER OF REPAIRMEN
''                 TO SATISFY THE CONSTRAINTS.
''OUTPUT:
''ROUTINE PRINTS THE OPTIMUM NUMBER OF REPAIRMEN (SERVERS) AND THE
''ASSOCIATED SYS PERFORMANCE STATISTICS INCLUDING STATE PROBABILITIES.
''
''P.NULL           PROBABILITY THE MAINTENANCE SYSTEM IS EMPTY.
''P.SYS.FULL       PROBABILITY THE MAINTENANCE SYSTEM IS FULL, I.E.,
''                 THE NUMBER OF CUSTOMERS IN SYS > OR = NSERVE.
''P.MACH.WAIT      PROBABILITY THAT AN ARRIVAL FOR MAINT MUST QUEUE.
''ENO.DOWN         EXPECTED (MEAN VALUE) NUMBER OF MACHINES DOWN.
''SDNO.DOWN        STANDARD DEVIATION OF NUMBER OF MACHINES DOWN.
''ENO.QUEUED       EXPECTED NUMBER OF MACHINES IN THE MAINT QUEUE.
''ESYS.WAIT        EXPECTED WAIT (MINUTES) IN THE MAINTENANCE SYSTEM.
''EQ.WAIT          EXPECTED WAIT (MINUTES) IN THE MAINTENANCE  QUEUE.
''EQ.WAIT.GQ       EXPECTED WAIT, GIVEN A MACHINE MUST WAIT FOR SERVICE.
''SDQ.WAIT.GQ      STD DEV  WAIT, GIVEN A MACHINE MUST WAIT FOR SERVICE.
''E.BUSY.SERVERS   MEAN NUMBER OF BUSY SERVERS.
''SD.BUSY.SERVERS  STD DEV NUMBER OF BUSY SERVERS.
''P.NSTATE         VECTOR IN WHICH THE N TH ELEMENT IS THE PROBABILITY
''                 THAT N MACHINES ARE DOWN.
''ENDOGENOUS VARIABLES:
''QV               VECTOR IN WHICH THE N TH ELEMENT IS THE PROBABILITY THAT
''                 AN ARRIVAL FOR SERVICE FINDS THE SYSTEM IN THE N TH STATE.
''Q.NULL           PROBABILITY THAT A MACHINE NEEDING SERVICE FINDS THE
''                 SYSTEM EMPTY.
''
DEFINE MODE AS A TEXT VARIABLE
DEFINE I, IPRINT, J, M, MAX, MPOP, N, NSERVE, AND N.TEN.PCT AS INTEGER VARIABLES
DEFINE P.NSTATE AND QV AS REAL, 1-DIMENSIONAL ARRAYS
LET R=LAMBDA/MU
LET N.TEN.PCT=MAX.F(1,INT.F(0.10*REAL.F(MPOP)))
```

```
RESERVE P.NSTAIL(*) AS MPOP
RESERVE QV(*) AS MPOP
''
''ITERATION OVER THE NUMBER SERVING STARTS HERE.
''
'S' LET C=NSERVE
    IF NSERVE GE MPOP
        RETURN
    OTHERWISE
LET RHO=R/C
LET ENO.DOWN=0.0
LET E.BUSY.SERVERS=0.0
LET VAR.BUSY.SERVERS=0.0
LET SUM=1.0
LET QSUM=C
LET NFACTORIAL=1.0
LET RATIO.FACTORIALS=1.0
IF NSERVE=1
    GO TO PASS
OTHERWISE
FOR N=1 TO NSERVE-1 DO
    LET NFACTORIAL=NFACTORIAL*N
    LET RATIO.FACTORIALS=RATIO.FACTORIALS*(MPOP-N+1)
    LET P.N=R**N/NFACTORIAL*RATIO.FACTORIALS ''OMITTING *P.NULL
    LET P.NSTATE(N)=P.N ''WITH THE NTH STATE STORED IN THE NTH ELEMENT
    ADD N*P.N TO ENO.DOWN
    ADD (NSERVE-N)*P.N TO QSUM
    ADD P.N TO SUM
    ADD N*P.N TO E.BUSY.SERVERS
    ADD N**2*P.N TO VAR.BUSY.SERVERS
LOOP ''TO DEVELOP SUMS
''
''CALCULATE THE LAST TERM IN THE SUM FOR P.NULL
''
'PASS'LET CFACTORIAL=C*NFACTORIAL
    LET CEXPOC=C**NSERVE
    LET COEF=CEXPOC/CFACTORIAL
    LET SUM.LAST=0.0 ''SUM FOR PROB THAT NO IN SYSTEM > OR = NSERVE
    LET VNO.DOWN=VAR.BUSY.SERVERS
FOR N=NSERVE TO MPOP DO
    LET RATIO.FACTORIALS=RATIO.FACTORIALS*(MPOP-N+1)
    LET P.N=COEF*RATIO.FACTORIALS*RHO**N
    ADD P.N TO SUM.LAST ''OMITTING *P.NULL
    ADD N*P.N TO ENO.DOWN ''OMITTING *P.NULL
    ADD N**2*P.N TO VNO.DOWN ''OMITTING *P.NULL
    LET P.NSTATE(N)=P.N ''OMITTING *P.NULL
LOOP ''TO OBTAIN THE LAST TERM
    ADD SUM.LAST TO SUM
    LET P.NULL=1.0/SUM ''PROB OF SYSTEM NULL STATE
''
''CHECK OF CALCULATED MAINTENANCE SYSTEM STATE-PROBABILITIES
''
    IF P.NULL GE 1.0 OR P.NULL LE 0.0
        PRINT 1 LINE WITH P.NULL THUS
ERROR IN CALCULATING STATE PROBABILITIES.  P.NULL = *.****.
        STOP
    OTHERWISE ''CALCULATE EXPECTED VALUES
```

```
      LET ENO.DOWN=ENO.DOWN*P.NULL
      LET VNO.DOWN=VNO.DOWN*P.NULL-ENO.DOWN**2
      LET SDNO.DOWN=SQRT.F(VNO.DOWN)
      LET P.SYS.FULL=P.NULL*SUM.LAST
      LET E.BUSY.SERVERS=P.NULL*E.BUSY.SERVERS+C*P.SYS.FULL
      LET VAR.BUSY.SERVERS=P.NULL*VAR.BUSY.SERVERS+C**2*P.SYS.FULL
    . -E.BUSY.SERVERS**2
      LET SD.BUSY.SERVERS=SQRT.F(VAR.BUSY.SERVERS)
      LET ENO.QUEUED=ENO.DOWN-NSERVE+P.NULL*QSUM
      LET ESYS.WAIT=ENO.DOWN/LAMBDA/(REAL.F(MPOP)-ENO.DOWN)
''ABOVE EXPRESSION IS KNOWN AS LITTLE'S FORMULA.
      LET EQ.WAIT=ESYS.WAIT-1.0/MU
''
''CALCULATE MAINTENANCE SYSTEM STATE PROBABILITIES AND THE PROBABILITY
''THAT THE NUMBER OF MACHINES DOWN EXCEEDS N.TEN.PCT.
''
      LET P10=0.0
FOR N=1 TO MPOP DO
      LET P.NSTATE(N)=P.NSTATE(N)*P.NULL
      IF N GE N.TEN.PCT
          ADD P.NSTATE(N) TO P10
      ALWAYS
LOOP ''OVER ALL STATES
''
''GO TO THE PRINT-OUTPUT SECTION IF THE MODE IN THIS ROUTINE IS  NOT "OPTIMIZE".
''
      IF MODE NE "OPTIMIZE"
          GO TO PRNT
      OTHERWISE ''TEST FOR CONSTRAINTS
      IF P10>P10LIM AND ENO.QUEUED/REAL.F(MPOP)>FQLIM
          ADD 1 TO NSERVE
          IF NSERVE > MPOP
              SUBTRACT 1 FROM NSERVE
              GO TO PRNT
          OTHERWISE
          GO TO S
      OTHERWISE ''PRINT OUTPUT
'PRNT' IF IPRINT=1
          START NEW PAGE
      ALWAYS
''
''CALCULATE THE NORMALIZATION CONSTANT FOR QV(*).
''
      LET NORM.CONST=MPOP*P.NULL
FOR N=1 TO MPOP-1 DO
      ADD (MPOP-N)*P.NSTATE(N) TO NORM.CONST
LOOP ''TO CALCULATE THE NORMALIZATION CONSTANT
''
''CALCULATE THE ELEMENTS OF QV(*).
''
      LET Q.NULL=MPOP*P.NULL/NORM.CONST
      LET EQ.WAIT.GQ=1.0
      LET VG.WAIT.GQ=0.0
FOR N=1 TO MPOP-1 DO
      LET QV(N)=(MPOP-N)*P.NSTATE(N)/NORM.CONST
      IF N GE NSERVE
          LET J=N-NSERVE+1
```

```
            ADD J*QV(N) TO EQ.WAIT.GQ
            ADD J*(J+1)*QV(N) TO VQ.WAIT.GQ
        ALWAYS
    LOOP ''TO GET QV(*)
        LET Q.SPACE=1.0-Q.NULL
        IF NSERVE=1
            GO TO 11
        OTHERWISE
    FOR N=1 TO NSERVE-1 DO
        SUBTRACT QV(N) FROM Q.SPACE
    LOOP ''TO FIND Q.SPACE
    '11'LET P.MACH.WAIT=Q.SPACE
        LET P.SYS.OPEN=1.0-Q.SPACE
        LET NMU=NSERVE*MU ''THE MAX SERVICE RATE FOR A FULL SYSTEM
        LET EQ.WAIT.CHECK=EQ.WAIT.GQ/NMU ''TO PROVIDE A CALCULATIONAL CHECK
        LET VQ.WAIT.CHECK=VQ.WAIT.GQ/NMU/NMU-EQ.WAIT.CHECK**2
        LET SDQ.WAIT.CHECK=SQRT.F(VQ.WAIT.CHECK)
    ''
    ''CALCULATE THE CONDITIONAL MEAN AND STANDARD DEVIATION OF THE
    ''WAITING TIME IN QUEUE, GIVEN A MACHINE MUST WAIT.
    ''
        LET EQ.WAIT.GQ=EQ.WAIT.GQ/NMU/P.MACH.WAIT
        LET VQ.WAIT.GQ=VQ.WAIT.GQ/NMU/NMU/P.MACH.WAIT-EQ.WAIT.GQ**2
        LET SDQ.WAIT.GQ=SQRT.F(VQ.WAIT.GQ)
        IF IPRINT NE 1
            RETURN
        OTHERWISE
        PRINT 4 LINES THUS

SERVICE SYSTEM PERFORMANCE STATISTICS USING A FINITE-STATE QUEUEING MODEL
---------------------------------------------------------------------------
        PRINT 3 LINES WITH MODE, MPOP, NSERVE, MPOP*LAMBDA, MU
        THUS
OUTPUT OF ROUTINE TO ********* MAINTENANCE FOR A POPULATION OF **** MACHINES
WITH ** SERVERS AND HAVING A MAX ARRIVAL RATE OF **.***** MAINTENANCE ACTIONS
PER MINUTE AND A CONSTANT SERVICE RATE OF **.***** ACTIONS PER MAN PER MINUTE.
        SKIP 2 LINES
        PRINT 15 LINES WITH P.SYS.FULL,Q.NULL,P.MACH.WAIT,N.TEN.PCT,PIO,ENO.DOWN,
        SDNO.DOWN, ENO.QUEUED, E.BUSY.SERVERS, SD.BUSY.SERVERS, ESYS.WAIT, EQ.WAIT,
        EQ.WAIT.GQ, SDQ.WAIT.GQ, SDQ.WAIT.CHECK
        THUS
PROBABILITY THAT THE MAINTENANCE SYSTEM IS FULL __________   *.****
PROBABILITY THAT A FAILED MACHINE FINDS SYSTEM EMPTY ________   *.****
PROBABILITY THAT A MACHINE MUST QUEUE FOR SERVICE __________   *.****
PROBABILITY THAT THE NO MACHINES DOWN > OR = *** __________   *.****
EXPECTED NUMBER OF MACHINES DOWN ___________________________  **.****
STD DEV  NUMBER OF MACHINES DOWN ___________________________  **.****
EXPECTED NUMBER OF MACHINES QUEUED _________________________  **.****
EXPECTED NUMBER OF BUSY SERVERS ____________________________  **.****
STD   DEVIATION OF BUSY SERVERS ____________________________  **.****
EXPECTED TIME (MIN) A MACHINE WAITS IN THE MAINT SYSTEM ____  ***.***
EXPECTED TIME (MIN) A MACHINE WAITS IN THE MAINT  QUEUE ____  ***.***
EXPECTED QUEUE TIME (MIN) GIVEN A QUEUE ____________________  ***.***
STD DEV  QUEUE TIME (MIN) GIVEN A QUEUE ____________________  ***.***
STD DEV  QUEUE TIME (MIN) UNCONDITIONALLY __________________  ***.***
```

```
        SKIP 2 LINES
        PRINT 7 LINES THUS
  MAINTENANCE SYSTEM STATE PROBABILITY DISTRIBUTIONS

-------------UNCONDITIONAL---------------GIVEN ARRIVAL--------------------------
NO MACH     PROB       CUML             PROB       CUML
IN  SYS     DENS       PROB             DENS       PROB
--------------------------------------------------------------------------------
        LET CUM.DIST=P.NULL
        LET CUM.DIST.COND=Q.NULL
        LET N=0
        PRINT 1 LINE WITH N, P.NULL, CUM.DIST, Q.NULL, CUM.DIST.COND
        THUS
     **     *.****     *.****           *.****     *.****
    FOR N=1 TO MPOP DO
        ADD P.NSTATE(N) TO CUM.DIST
        ADD QV(N) TO CUM.DIST.COND
        PRINT 1 LINE WITH N, P.NSTATE(N), CUM.DIST, QV(N), CUM.DIST.COND
        THUS
     **     *.****     *.****           *.****     *.****
    LOOP ''OVER THE SYSTEM STATES
        PRINT 2 LINES THUS
---------------------------------------------------------------------------------

        LET DELT=MAX.F(1.0,TRUNC.F(EQ.WAIT.GQ/10.0))
        LET MAX=60
        SKIP 2 LINES
    ''
    ''PRINT HEADINGS FOR QUEUE WAITING TIME DISTRIBUTION.
    ''
        PRINT 7 LINES WITH Q.NULL THUS
CUMULATIVE AND CONDITIONAL CUM PROB DISTRIBUTIONS OF WAITING TIME IN QUEUE

----------------------------------------------------------------------------------
 TIME     CUML     COND
 (MIN)    PROB     PROB
----------------------------------------------------------------------------------
 0       *.****     0
    ''
    ''START TIME LOOP FOR WAITING TIME DISTRIBUTIONS.
    ''
    FOR I=1 TO MAX ''TIME STEPS'' DO
        LET TM=I*DELT
        LET ARG=TM*NMU
        LET CUM.DIST=1.0-QV(NSERVE)*EXP.F(-ARG)
        LET SUMN=0.0
        IF MPOP-NSERVE<2
           GO TO BYN
        OTHERWISE
        FOR N=NSERVE+1 TO MPOP-1 DO
            LET SUMJ=1.0
            LET ARGJ=1.0
            LET JFACTORIAL=1.0
            FOR J=1 TO N-NSERVE DO
                LET ARGJ=ARGJ*ARG
                LET JFACTORIAL=JFACTORIAL*J
```

```
ROUTINE BEST.SERVICE                 CACI SIMSCRIPT II.5 for PRIME Systems, Release 2.1        Page   29
Options = SEQUENCE,ID,SUBCHK,XREF,NOEXPLIST,TRACE3                   06 JAN 1983 15:25:50

  245             ADD ARGJ/JFACTORIAL TO SUMJ
  246          LOOP ''OVER J
  247          ADD GV(N)*SUMJ TO SUMN
  248       LOOP ''OVER N
  249  'BYN' LET CUM.DIST=CUM.DIST-EXP.F(-ARG)*SUMN
  250       LET CON.DIST=(CUM.DIST-P.SYS.OPEN)/P.MACH.WAIT
  251       PRINT 1 LINE WITH TM, CUM.DIST, AND CON.DIST THUS
****.**     *.****    *.****
  252    LOOP ''OVER I TIME STEPS
  253       PRINT 2 LINES THUS
-----------------------------------------------------------------------------------
  254       RETURN
  255       END ''OF BEST.SERVICE

                         C R O S S - R E F E R E N C E

 NAME                        TYPE                                 MODE      LINE NUMBERS OF REFERENCES

ARG                    RECURSIVE VARIABLE   WORD  78    DOUBLE     232    233    243    249
ARGJ                   RECURSIVE VARIABLE   WORD  84    DOUBLE     240    243*   245
BEST.SERVICE           ROUTINE                          INTEGER      1
BYN                    LABEL                                       236    249
C                      RECURSIVE VARIABLE   WORD  10    DOUBLE      63     67     72     92     93    118    119
CEXPOC                 RECURSIVE VARIABLE   WORD  28    DOUBLE      93     94
CFACTORIAL             RECURSIVE VARIABLE   WORD  26    DOUBLE      92     94
COEF                   RECURSIVE VARIABLE   WORD  30    DOUBLE      94     99
CON.DIST               RECURSIVE VARIABLE   WORD  88    DOUBLE     250    251
CUM.DIST               RECURSIVE VARIABLE   WORD  69    DOUBLE     208    211    214*   216    233    249*   250
                                                                   251
CUM.DIST.COND          RECURSIVE VARIABLE   WORD  71    DOUBLE     209    211    215*   216
DELT                   RECURSIVE VARIABLE   WORD  73    DOUBLE     220    231
E.BUSY.SERVERS         ARGUMENT             NO.   19    DOUBLE       3     69     86*   118*   120    202
ENO.DOWN               ARGUMENT             NO.   12    DOUBLE       2     68     83*   101*   114*   115    122
                                                                   123*   202
ENO.QUEUED             ARGUMENT             NO.   14    DOUBLE       3    122    143    202
EQ.WAIT                ARGUMENT             NO.   16    DOUBLE       3    125    202
EQ.WAIT.CHECK          RECURSIVE VARIABLE   WORD  61    DOUBLE     185    186
EQ.WAIT.GO             ARGUMENT             NO.   17    DOUBLE       3    165    171*   185    192*   193    202
                                                                   220
ESYS.WAIT              ARGUMENT             NO.   15    DOUBLE       3    123    125    202
EXP.F                  ROUTINE                          DOUBLE     233    249
FQLIM                  ARGUMENT             NO.    6    DOUBLE       1    143
I                      RECURSIVE VARIABLE   WORD   1    INTEGER     54    230*   231
II                     LABEL                                       177    182
INT.F                  ROUTINE                          INTEGER     57
IPRINT                 ARGUMENT             NO.    7    INTEGER      1     54    151    195
J                      RECURSIVE VARIABLE   WORD   2    INTEGER     54    170    171    172*   242*   244
JFACTORIAL             RECURSIVE VARIABLE   WORD  86    DOUBLE     241    244*   245
LAMBDA                 ARGUMENT             NO.    1    DOUBLE       1     56    123    199
M                      RECURSIVE VARIABLE   WORD   3    INTEGER     54
MAX                    RECURSIVE VARIABLE   WORD   4    INTEGER     54    221    230
MAX.F                  ROUTINE                          INTEGER     57    220
MODE                   ARGUMENT             NO.    8    TEXT         2     53    140    199
MPOP                   ARGUMENT             NO.    3    INTEGER      1     54     57     58     59     64     80
                                                                    97     98    123    131    143    145    157
```

```
                                                                     158    159    164    167    168    199*   213
                                                                     235    238
MU                  ARGUMENT              NO.     2        DOUBLE      1     56    125    184    199
N                   RECURSIVE VARIABLE    WORD    5        INTEGER    54     78*    79     80     81     82     83
                                                                      84     86     87     97*    98     99    101
                                                                     102    103    131*   132*   133    134    158*
                                                                     159*   167*   168*   169    170    171    172
                                                                     179*   180    210    211    213*   214    215
                                                                     216*   238*   242    247
N.TEN.PCT           RECURSIVE VARIABLE    WORD    6        INTEGER    54     57    133    202
NFACTORIAL          RECURSIVE VARIABLE    WORD   20        DOUBLE     73     79*    81     92
NMU                 RECURSIVE VARIABLE    WORD   59        DOUBLE    184    185    186*   192    193*   232
NORM.CONST          RECURSIVE VARIABLE    WORD   49        DOUBLE    157    159*   164    168
NSERVE              ARGUMENT              NO.     4        INTEGER     1     54     63     64     75     78     84
                                                                      93     97    122    144*   145    146*   169
                                                                     170    176    179    184    199    233    235
                                                                     238    242
P.MACH.WAIT         ARGUMENT              NO.    11        DOUBLE      2    182    192    193    202    250
P.N                 RECURSIVE VARIABLE    WORD   24        DOUBLE     81     82     83     84     85     86     87
                                                                      99    100    101    102    103
P.NSTATE            ARGUMENT              NO.    21  (1-D) DOUBLE      4     55     58*    82    103    132*   134
                                                                     159    168    214    216
P.NULL              ARGUMENT              NO.     9        DOUBLE      2    106    110*   111    114    115    117
                                                                     118    119    122    132    157    164    208
                                                                     211
P.SYS.FULL          ARGUMENT              NO.    10        DOUBLE      2    117    118    119    202
P.SYS.OPEN          RECURSIVE VARIABLE    WORD   57        DOUBLE    183    250
P10                 RECURSIVE VARIABLE    WORD   47        DOUBLE    130    134*   143    202
P10LIM              ARGUMENT              NO.     5        DOUBLE      1    143
PASS                LABEL                                             76     92
PRNT                LABEL                                            141    147    151
Q.NULL              RECURSIVE VARIABLE    WORD   51        DOUBLE    164    175    202    209    211    226
Q.SPACE             RECURSIVE VARIABLE    WORD   55        DOUBLE    175    180*   182    183
QSUM                RECURSIVE VARIABLE    WORD   18        DOUBLE     72     84*   122
QV                  RECURSIVE VARIABLE    WORD    7  (1-D) DOUBLE     55     59*   168    171    172    180    215
                                                                     216    233    247
R                   RECURSIVE VARIABLE    WORD    8        DOUBLE     56     67     81
RATIO.FACTORIALS    RECURSIVE VARIABLE    WORD   22        DOUBLE     74     80*    81     98*    99
REAL.F              ROUTINE                                INTEGER    57    123    143
RHO                 RECURSIVE VARIABLE    WORD   12        DOUBLE     67     99
S                   LABEL                                             63    149
SD.BUSY.SERVERS     ARGUMENT              NO.    20        DOUBLE      4    121    202
SDNO.DOWN           ARGUMENT              NO.    13        DOUBLE      2    116    202
SDQ.WAIT.CHECK      RECURSIVE VARIABLE    WORD   65        DOUBLE    187    202
SDQ.WAIT.GG         ARGUMENT              NO.    18        DOUBLE      3    194    202
SQRT.F              ROUTINE                                DOUBLE    116    121    187    194
SUM                 RECURSIVE VARIABLE    WORD   16        DOUBLE     71     85*   105*   106
SUM.LAST            RECURSIVE VARIABLE    WORD   32        DOUBLE     95    106*   105    117
SUMJ                RECURSIVE VARIABLE    WORD   82        DOUBLE    239    245*   247
SUMN                RECURSIVE VARIABLE    WORD   80        DOUBLE    234    247*   249
TM                  RECURSIVE VARIABLE    WORD   76        DOUBLE    231    232    251
TRUNC.F             ROUTINE                                INTEGER   220
UIB.W               IMPLIED SUBSCRIPT     SYS     4        INTEGER   201    206    222
VAR.BUSY.SERVERS    RECURSIVE VARIABLE    WORD   14        DOUBLE     70     87*    96    119*   121
VNO.DOWN            RECURSIVE VARIABLE    WORD   34        DOUBLE     96    102*   115*   116
VQ.WAIT.CHECK       RECURSIVE VARIABLE    WORD   63        DOUBLE    186    187
VQ.WAIT.GG          RECURSIVE VARIABLE    WORD   53        DOUBLE    166    172*   186    193*   194
```

ANNEX B

HETEROGENEOUS MACHINE POPULATION

The theory presented in the body of this report treats the case of a homogeneous population of machines (customers), whose reliability and maintainability (RAM) parameters are the same for all machines. Thus, equations (1), (2), and (3) do not distinguish between categories or types of machines having possibly different, intrinsic mean times between failure (MTBF) and mean times to repair (MTTR). For this ideal case, the approach presented in the body of the report is exact.

In many instances the population of machines is heterogeneous. In this case an approximation is needed for the RAM parameters which are equivalent to their homogeneous counterparts. This annex discusses an approximation which can be used in conjunction with the previous analytical results to yield queueing statistics which are in good agreement with simulated results for tandem production operations.

On a production line different machine operations may each have several machines of a common type. However, each type of machine (or operation) may have RAM parameters different from parameters of other types. Furthermore, in this setting the machines in a particular operation (n) do not operate continuously. Their failure behavior will reflect this condition. Assuming that the machine operations are asynchronous, a machine must wait: (a) if the downstream buffer is full, (b) if the upstream buffer is empty, or (c) if the machine is down for repairs. The definition of arrival rate per machine (λ) requires the use of the recriprocal of the actual--as opposed to intrinsic--mean clock time between failures. Call this the effective mean time between failures MTBF*. Similarly, the effective mean time to repair (MTTR*) must reflect the diversity of types of machines repaired, the MTTR for each type, and the relative demand for repair by type. Then, the repair parameter (μ_1) used in homogeneous queueing theory is the reciprocal of MTTR*.

The following method is a partial (and imperfect) attempt to account for the non-operating time, in addition to maintenance downtime, which the machines of a particular operation experience. Generally, the production rates of acceptable parts from the machine operations are not identical, i.e., the line

is unbalanced. The machines of an operation which does not have minimal thruput must wait a portion of the time. I define the operational ratio of operation n as the fraction of the average available machine-time which machines at this operation must work in order to meet line thruput. Reference is made to Table B-1 for a complete list of symbols and terms used in this discussion. The average number of machine-minutes actually worked during a day by machines at operation n is, then,

$$U(n) D\, A(n) \rho(n) \ , \tag{B-1}$$

where the symbols U, D, A, and ρ are defined in Table B-1. The expected number of maintenance actions required at this operation is this total operating time divided by the MTBF(n). This number is denoted $\nu(n)$. The total of average daily maintenance actions required is denoted by X. Formally,

$$X = \Sigma_{n=1}^{N} \nu(n) \quad . \tag{B-2}$$

The total daily average maintenance time required is the following sum over all operations:

$$T = \Sigma_{n=1}^{N} \nu(n) \mathrm{MTTR}(n) \quad . \tag{B-3}$$

The effective mean time, per machine, between failures is the maximum daily machine-minutes (D M) divided by X:

$$\mathrm{MTBF}^* = D\, M/X \quad . \tag{B-4}$$

The effective mean time to repair is

$$\mathrm{MTTR}^* = T/X \quad . \tag{B-5}$$

The equivalent arrival rate of maintenance actions--1/MTBF*--is the sum of required actions per unit time over all machine types divided by the total number of machines. The equivalent mean time to repair, or reciprocal equivalent service rate, is a weighted sum of MTTR's for each machine type. The weights in this expression represent the fraction of the total maintenance actions contributed by machines of this (n) type.

At this point the reader may have noticed that one of the outputs of the queueing model (W_q) is needed to define the equivalent values of λ and μ. Availability (A) of machines of a given type in a manufacturing operation is used to estimate the thruput of that operation. This, in turn, is used to develop parameters λ and μ of the queueing model. However, A must consider

the expected waiting time in the maintenance queue (W_q) as well as the intrinsic RAM characteristics of that type of machine. Since A involves W_q, which is not yet known, an iterative solution technique is required. This process starts by assuming that W_q is zero, calculates A for each machine type, which leads to estimates of λ and μ. Use of λ and μ in the queueing model, then, yields an improved estimate of W_q. The better estimate of W_q permits better estimates of λ and μ, and so forth. To accelerate the conversion of this iterative process, it is helpful to use a value of W_q (the same for all machines) which is the average of the i th and (i-1) th iterates. Convergence in some instances is not uniform. Computational experience indicated that truncation of this iterative process occurs rapidly using the following convergence criterion: the relative difference in values of W_q for the last two iterations must be less than 2%. Sound computer coding practice provides a sure escape from the iteration loop after a maximum number of iterations has occurred. In the TANDEMT program, the maximum was set to 20, altho iterative truncation has yet to occur.

TABLE B-1

NOTATION USED IN OBTAINING
EFFECTIVE ARRIVAL AND SERVICE RATES

Symbol	Description
D	length of workday (minutes)
U(n)	the number of (units of) machines of a common type in operation n
N	the number of machine types or machine operations
M	the total number of machine units , $= \Sigma_n^N U(n)$
MTBF(n)	the mean operating time between failures (minutes) of machines of type n
MTTR(n)	the mean time (minutes) to repair a machine of type n, given maintenance resources
WQ	the mean time (minutes) a machine spends in the repair queue
A(n)	the steady-state availability of machines of type n, $= MTBF(n)/[MTBF(n)+MTTR(n)+WQ]$
R(n)	unit machine rate (per min) at operation n
AP(n)	average daily production capacity at operation n $= U(n)D\ R(n)A(n)$
$\alpha(n)$	acceptance rate of parts from machines at operation n only
C(n)	cumulative acceptance rate of parts passing operation n per part starting the first operation $= \Pi_{j=1}^{n} \alpha(j)$
CP(n)	average daily production capacity of <u>acceptable</u> parts at operation n, $= AP(n)\ \alpha(n)$
n*	value of n for which CP(n) is a minimum $CP(n^*) \leq CP(n) \quad , \quad 1 \leq n \leq N$
RP(n)	average daily parts production at operation n required to meet the average daily line thruput, $= AP(n^*)\ C(n)/C(n^*)$

TABLE B-1 (Cont'd)

NOTATION USED IN OBTAINING
EFFECTIVE ARRIVAL AND SERVICE RATES

Symbol	Description
$\rho(n)$	operational ratio for machine operation n, = RP(n)/AP(n)
$\nu(n)$	average daily number of maintenance actions for machines of common type in operation n, = $U(n)D\ A(n)\rho(n)/MTBF(n)$
X	total average daily maintenance actions required, $= \Sigma_n^N \nu(n)$
T	total average daily maintenance time (min) required, $= \Sigma_n^N \nu(n)MTTR(n)$
MTBF*	effective mean time (min) between failures, = D M/X
MTTR*	effective mean time (min) to repair all machines, = T/X
λ^*	effective arrival rate (1/minute), = 1/MTBF*
μ_1^*	effective service rate (1/minute), = 1/MTTR*